AFRL-IF-RS-TR-2004-209
Final Technical Report
July 2004

ADVANCED CAPABILITIES FOR EVIDENCE EXTRACTION (ACEE)

Center for Natural Language Processing, Syracuse University

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-209 has been reviewed and is approved for publication

APPROVED: /s/
WALTER V. GADZ
Project Engineer

FOR THE DIRECTOR: /s/
JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** July 2004	**3. REPORT TYPE AND DATES COVERED** **FINAL Sep 01 – Feb 04**

4. TITLE AND SUBTITLE

ADVANCE CAPABILITIES FOR EVIDENCE EXTRACTION (ACEE)

5. FUNDING NUMBERS

G - F30602-01-2-0568
PE - 62301E
PR - EELD
TA - 01
WU - 14

6. AUTHOR(S)

Elizabeth Liddy, Nancy McCracken, Eileen Allen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Center for Natural Language Processing
Syracuse University
4-206 Center for Science and Technology
Syracuse NY 13244

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRLIF-RS-TR-2004-209

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Walter V. Gadz/IFEA/(315) 330-3948 Walter.Gadz@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

The Center for Natural Language Processing (CNLP) at Syracuse University recently completed the Advanced Capabilities for Evidence Extraction (ACEE) Project, which has improved the effectiveness of its basic entity, relation, and event extraction technology and extended these capabilities in several ways. First, the IE technology can now be quickly ported to new domains by use of algorithms utilizing Transformation-Based Learning to specialize generic relation extraction to specific domains. Second, Alias Tracking has enhanced entity coalition within documents by means of more sophisticated co-reference algorithms and has enabled entity tracking across documents. Third, new Linguistic Inferencing capabilities improved cohesiveness of extractions by enabling event coalition. Fourth, significant development in Temporal Sequencing and Scenario Understanding was accomplished based on improved interpretation of temporal attributes of events (e.g., frequency, occurs, since) and temporal relations between events (e.g., before, after, concurrent) providing a richer basis for timeline analysis of events than in previous extraction work. Fifth, an innovative model-based approach to automated certainty detection and categorization was developed and tested, including level of certainty, the experiencer of certainty (e.g., reporter, witness), the abstract or factual nature of the focus of the certainty, and point of time at which the certainty is expressed. Engineering efforts have improved the speed, scalability, and portability of CNLP's IE technology.

14. SUBJECT TERMS

Evidence Extraction, Relation Extraction, IE, Natural Language Processing, Transformation-Based Learning, Temporal Extraction, Certainty Model

15. NUMBER OF PAGES

56

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

List of Figures

I. Introduction

The Advanced Capabilities for Evidence Extraction project has successfully developed a suite of rich extraction capabilities for recognizing, interpreting and representing the entities, events and relations from text. This 30 month research project was completed by the Center for Natural Language Processing (CNLP) at Syracuse University under the Evidence Extraction and Link Discovery (EELD) Program. The goal of the Advanced Capabilities for Evidence Extraction Project was to enable the down-stream Link Discovery (LD) and Pattern Learning (PL) modules to accomplish their goals with the broadest coverage and highest accuracy for alerting US national security agencies to impending asymmetric threats.

The evidence extraction capabilities were developed for this project under six research areas: Transformation-Based Learning for Specific Domains; Alias Tracking; Temporal Sequencing; Linguistic Inferencing for Event Coreference; Confidence Levels based on Linguistic Certainty; and Temporal Extractions for Scenarios. In addition to these six research areas, work on the project included evaluation, participation in the Automatic Content Extraction (ACE) program and general improvements to the extraction system.

Prior to the EELD program, CNLP's document processing capabilities included the generic extraction from text of events and entities, and the ability to specialize those extractions to a domain, enabling applications such as Question/Answering (QA) and Visualization using the entities and events in the taxonomy of the user's specific domain. Some of this capability was developed under the prior EELD Seedling Project. The research areas carried out under the EELD project enhanced those capabilities, firstly by using the machine learning technique of Transformation Based Learning (TBL) to reduce the time needed to move to a new specific domain, and secondly by improving and adding to the relations that are extracted. This included using Alias Tracking and Inference to coalesce entities and events at the discourse level, and adding attributes to extractions representing Confidence Levels, as well as two aspects of temporal information in Temporal Sequencing and Scenarios. All of these capabilities, in addition to general improvements in relation extraction, enable a rich set of transactional, social, temporal and geographical relationships to be derived from the extractions and put into a database of relations for use by Link Discovery applications.

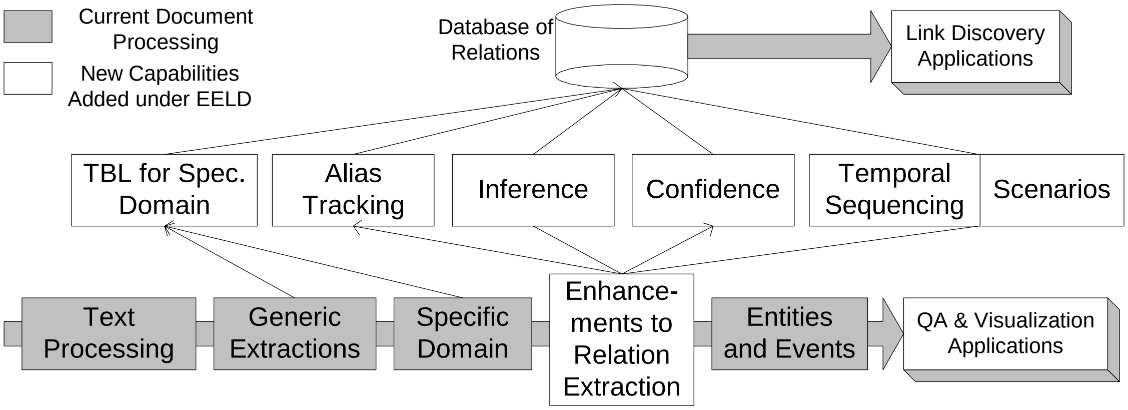


Figure 1: Current (baseline) document processing capabilities enhanced under the EELD program

The remainder of this report contains a description of the baseline CNLP extraction capability, part of which was developed under the EELD Seedling Project, followed by reports on the areas in which research was conducted in this project. The two related research areas of Temporal Sequencing and Scenarios are combined in one section, yielding five sections on research. This is followed by a section describing the generic extraction improvement that was carried out throughout this project and the evaluations that were conducted, both internal and program-wide. The final sections describe software development and deliveries, transitions of the software to other organizations, both within and outside the EELD program, and research papers and presentations that were produced by CNLP under the EELD program or in related research.

II. Baseline Extraction Capability

The extraction capabilities developed for this project were part of the eQuery document processing system. This system was developed as a primarily rule-based system to identify entities in the text. In the EELD Seedling project, a generic event-based extraction capability was added to the document processing.

The document processing system begins with the entity identification phases:

1. part-of-speech tagging
2. detection of non-compositional phrases, which are linked as single concepts
3. identification of numeric-concept phrases, named entities, and complex nominal phrases
4. categorization of named entities and numeric concepts

In each of these phases, information is added to the text, for example, in the sentence:

Colonel Khaddafi gave a two hour speech.

the text would be marked up as follows with the two basic entities in the sentence:

<S> <NP cat="person"> Colonel|NP_Khaddafi|NP </NP> gave|VBD a|DT <CN> <NC cat=”time”> two|CD hour|NN </NC> speech|NN </CN> .|. </S>

The extraction process uses this marked-up text to identify events and relations. This information extraction process is done in two parts. The first part is generic extraction, where all events and entities are extracted in an open domain mode, using abstract role names, such as *agent* and *object*. The second stage maps the generic extractions to a specific subject domain, based on a model of the domain. This model includes a lexicon of terminology for that domain and a mapping for event verbs (and nominalizations) and their roles. Note that in the EELD terminology the event semantic roles are relations between events and entities.

One advantage of this two-step extraction procedure is that the generic extraction rules, which are based on the more syntactic types of information in English sentences, are developed only once, instead of developing new rules for every domain. Another advantage is that generic extraction has the capability of always extracting a more complete set of information, since it will give extractions for all verbs as events and all types of entities, not just those it recognizes as specialized entities and events in the domain of interest.

In the generic extraction process, events and entities are extracted at the sentence level by means of a set of rules that specify sentence grammar patterns and are labeled with generic role names, as suggested in Case Grammar (Fillmore, 1968). The extraction rules are expressed in a rule language designed at CNLP, in which the rules can be efficiently implemented using regular expressions. This style of processing is also referred to as shallow parsing, as it is more efficient than a full deep parser, such as (Collins, 1996).

The events in the generic extraction system are based on semantic verb classes from an abstract case grammar model. This model is similar in its level of abstraction and in its choice of case roles to (Cook 1998), whose case role model is based on Fillmore (1968) and others. The case frames capture the relationships between events and entities that exist at a semantic, conceptual level, regardless of the surface syntactic structure. Case roles may be missing from the text due to English language constructs that allow deleted roles, co-referential roles and lexicalized roles.

In developing a set of case role labels for the case frames, an initial set of generic case role labels was taken primarily from Sowa's conceptual graph relations (Sowa, 1984), where the relations can be interpreted as case roles. This model was refined during the process of writing the extraction rules to reflect a model based on what can be extracted from the text. Note that these case role labels are more general than the *proposition argument roles* or *semantic roles* developed in Propbank (Kingsbury 2002) and FrameNet (Baker, 1998). The CNLP case role labels are included in the taxonomy given in Appendix A.

In addition to the generic event extraction, a set of rules has been developed for entity relation and attribute extraction. These also operate at the sentence level to extract attributes of entities, such as *title* for person entities, and entity relations, such as *spouse* between two person entities or *employer* between an organization entity and a person entity.

For each document, the collection of entities, events and relations are saved in an extraction data structure. The entity and event extractions are represented as frames, and attributes and relations are represented as slots in the frames.

Consider the example sentence:
Colonel Kaddafi paid Carlos Alhaddin two thousand dollars on March 15, 1996.

The main extractions from this sentence would be two entities and an event:

```
id = 0
named_entity = Kaddafi
type = person
title = Colonel
```

```
id = 1
named_entity = Carlos Alhaddin
type = person
```

```
id = 2
event = pay
        agent = Colonel Kaddafi = id 0
        object = Carlos Alhaddin = id 1
        amount = two thousand dollars
        occurs = March 15, 1996
```

In this simple one sentence example, the frame for the event "pay" has a set of slots filled in with attributes and relations about this event.

For the next step, the extractions are specialized to a specific domain. For any specific domain, the generic case model is mapped to a refinement model based on the domain. This model breaks out the generic classes of verbs to more specific semantic verb classes based on the domain. The model then specifies how to map the generic case role names for specific verb classes to domain-specific case role names. This process may also involve some cases in which roles are added or coalesced to fit the domain-specific verb case model.

Example of specific domain case roles:
event = pay
buyer = Colonel Kaddafi = id 0
seller = Carlos Alhaddin = id 1
payment = two thousand dollars
occurs = March 15, 1996

When the extraction processing of a single document is complete, the extractions from the document are formatted in an XML structure with document level information. This document information includes the document date and ID.

III. Research Areas

III.A Transformation-Based Learning of Specific Domains:

In the early days of Information Extraction (IE), IE systems extracted a limited amount of information in order to fill in the blanks in a predefined domain-specific template. The problem with these systems was their inability to be easily ported to new domains. The generic extraction model is a significant expansion in the ability to capture a wide variety of useful information in various domains; the remaining issue is to develop a method to easily port the extraction capability to new domains.

In this research area for the EELD project, we used Transformation Based Learning (TBL) to learn domain-specific specializations for generic event extractions. The primary goal of this task was to use machine learning to reduce the amount of human effort required for specializing generic event extractions to new domains.

Description of the event specialization task

The task of learning specialization of generic event extractions involves varying levels of complexity. At the simplest level, an event type is learned and the generic roles are relabeled to more specific ones.

For example, in the sentence,

> *Just today, a business near the French Embassy was blown up and four people were killed by an unidentified gunman.*

the two extracted events can be easily labeled.

event = kill *agent = unidentified gunman* *object = four people*	*event = blow_up* *object = business*

After specialization, the event representation becomes

event = kill *type = kill* *perpetrator = unidentified gunman* *victim = four people*	*event = blow_up* *type = attempt-to-kill* *victim = business*

However, even this simpler form of event specialization can be complicated by word sense ambiguity. For example, not all instances of the event “pay” should be specialized; if the generic indirect object is “attention” (pay attention) or “respects” (pay respects), then the event should not be specialized as this sense of the word “pay” falls outside of the domain model that includes the payment event.

A greater level of complexity is seen when an event frame is restructured to more appropriately capture the conceptual meaning for a specific domain. This can happen when the more specific role has particular semantic patterns or when some senses of the verb have different semantic patterns. As an example of the latter, in general, the verb “commit” may take an object role, but in this domain, the syntactic object may actually be the event.

> *But in November Rachuk committed suicide in Russia, and it was some time later that a certain Sadykov asked the bosses of Summit International for a meeting.*

event = commit *agent = Rachuk* *object = suicide* *location = Russia* *occurs = November*

This frame is re-structured to make “suicide” be the event.

event = commit suicide *type = kill* *agent = Rachuk* *location = Russia* *occurs = November*

Transformation-Based Error-Driven Learning (TBL)

TBL is a robust corpus-based machine learning paradigm that is comprised of unannotated text, an initial state annotator that can be at any level of sophistication (Brill, 1993) but is typically based on a naïve and simplistic algorithm, a hand-tagged or hand-corrected annotated corpus considered to be the gold standard, a set of transformation templates, and an iterative learning program which learns the transformation rules necessary to change the annotations of the initial state annotator to match those found in the gold standard corpus. This approach is error-driven because the transformations learned at each step of the iteration are those that lead to the greatest reduction in errors when compared to the gold standard. The transformation-based error-driven learning paradigm is illustrated in Figure 2, adapted from (Ramshaw & Marcus, 1996b).

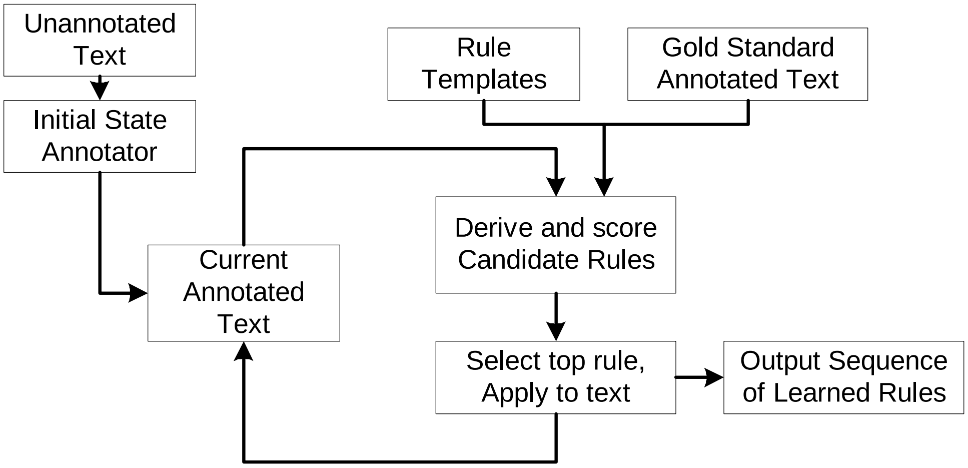

Unannotated Text
Initial State Annotator
Rule Templates
Gold Standard Annotated Text
Derive and score Candidate Rules
Current Annotated Text
Select top rule, Apply to text
Output Sequence of Learned Rules

Figure 2: Transformation-Based Error-Driven Learning Paradigm

After the initial annotation of the text, TBL has a loop for the main learning process. It consists of using the rule templates and the current annotated text to derive all possible instances of transformation rules that can transform an instance of the text. To score the rules, each rule is applied to the entire text and the result is compared to the gold standard to obtain the number of "errors" between the two. The rule which minimizes this error is selected and applied to the text to obtain a new current annotated text. The selected rule is also added to the output rules, and the process continues until errors can no longer be reduced.

Transformation-based, error-driven learning has the following advantages: (a) a relatively small number of rules that are linguistically motivated and understandable to both humans and machines are created, (b) a wide range of symbolic vs. statistical linguistic regularities are exploited, (c) an initial annotation is iteratively transformed into one with fewer errors, (d) an order of magnitude fewer decisions are required compared to estimating the parameters of statistical models, (e) it is resistant to overtraining (Ramshaw & Marcus, 1996a), and (f) more powerful than decision trees.

Transformation-based error-driven learning has been successfully applied to numerous NLP tasks, including learning rules for part-of-speech tagging (Brill, 1993; Brill, 1994; Brill, 1995); prepositional phrase attachment (Brill & Resnik, 1994; Yeh & Vilain, 1998); subordinate conjunction attachment (Yeh & Vilain, 1998); parsing (Brill, 1993; Satta & Brill, 1996); word segmentation (Palmer, 1997; Hockenmaier & Brew, 1998); and grammatical relation extraction (Ferro, Vilain, & Yeh, 1999).

For EELD, the fnTBL transformation-based learning software package[1] from Johns Hopkins University was used for machine learning. This software is free, open source software with no licensing issues for its use in a system. It is a flexible package that allows a variety of natural language processing tasks to be specified by defining a set of templates that establish the context of information to be learned. The fnTBL software package has also been developed to significantly speed up the learning process over earlier TBL systems.

In setting up the learning problem, the most important part is defining the templates. The templates define what context is used in identifying training instances, and is equivalent to defining the feature set in other machine learning methods. The templates are patterns for rules that have two parts: the first part identifies the triggering environment, or the features, and the second part identifies a transformation, or re-write rule on one of the features. An example of a template for this task could be paraphrased as:

> Given an extraction text with a slot that has slotname, slotvalue and the type of the slotvalue, relabel the slotname

The template rules are used in order during the learning process, and are organized so that more general rules are learned first. A more general rule template example than the above one, could apply to any slotvalue that had a particular type. A sample transformation rule that could be generated and learned from this template is:

> For any event frame with the text “kill”, if the “object” slot has type “person”, relabel it to be “victim”.

These templates were set up in an experimental process that used exemplar sentences, described in the next section, to design the order and generality of the learned rules.

Another issue in setting up TBL learning tasks is to set up a threshold value for the scoring function. The scoring function is based on taking the number of good rule applications, where the error is reduced, and subtracting the number of bad rule applications, where the error is increased. In the fnTBL toolkit, the threshold value indicates a learning cutoff such that the net gain in learning must be one more than the value of the threshold. The studies by Ramshaw and Marcus (1996) showed that TBL systems do not overtrain if the threshold is set so that the difference between good and

[1] The fnTBL Toolkit web page is at http://nlp.cs.jhu.edu/~rflorian/fntbl/.

bad rule applications is at least more than one. In the case of sparse data, this means that at least two examples of any text pattern must be in the training data for a rule to be learned.

Training and Testing

The data used for this study came from the Russian Contract Killing corpus supplied for the EELD project. Since the main goal was to reduce the human effort to move to a new domain, the gold standard training data was developed in several annotation cycles using bootstrapping. Each cycle consisted of the human analyst marking system output data to serve as the gold standard, training a set of transformation rules using TBL, and then rerunning the system using these transformation rules to produce specialized events, followed by evaluation. Further annotation could then take place on system output that was already transformed.

The first annotation cycle was purposefully small in order to define the templates. It was based on 14 exemplar sentences that were hand-picked by the analysts as representative of the domain-specific events that they wished to specialize; these sentences contained examples of such domain-specific events as *kill*, *murder* and *apprehend* among others. The analysts provided the generic extractions from these sentences along with the specialized extractions for these sentences. Using the generic extractions as the baseline and the matching specialized extractions as the gold standard; from these first 14 sentences, 26 specialization rules were learned using the fnTBL toolkit and a learning threshold of 0. A value of zero for the threshold was used to bootstrap the gold standard annotations.

For the second annotation cycle, 35 documents were chosen from the available corpus using a frequency count of the desired events.

Event Type	**Associated Events**
arrest	arrest
detain	apprehend, detain
kidnap	abduct, kidnap
kill	assassinate, execute, kill, murder

Figure 3: Event Types and Associated Events for 2nd Annotation Cycle

The initial set of 26 transform rules were applied to this set of 35 documents to bootstrap the annotation for the gold standard. Then the documents, along with both the generic extractions and the bootstrapped transformed extractions, were provided to the analysts for hand-correction of the preliminary transformed extractions. During this hand-correction effort, the analysts added event types where they were needed, specialized slot names where applicable, restructured event frames where necessary, and corrected the generic event extraction errors. The hand-correction effort was completed in 8 hours. The generic extractions and matching gold standard extractions from this set of 35 documents were fed into the fnTBL toolkit; again using threshold 0, and 272 transform rules were produced.

For the third annotation cycle, another 35 documents were chosen based on a larger set of events of interest:

Event Type	**Associated Events**
arrest	arrest, charge
attempt-to-kill	attempt to assassinate, attempt to kill, attempt to murder
deal	bargain
detain	apprehend, detain, extradite
disappear	disappear, escape
investigation	interrogate, investigate, probe
kidnap	abduct, kidnap
kill	assassinate, execute, kill, murder
payment	invest, pay
telephone conversation	answer phone
theft	steal, theft
warn	alert, warn

Figure 4: Event Types and Associated Events for 3rd Annotation Cycle

As in the second annotation cycle, the rules learned from the prior cycle (in this case 272 rules) were applied to this set of 35 documents, once again in an effort to bootstrap the annotation of the gold standard. The resulting set of documents, generic extractions and preliminary specialized extractions was turned over to the analysts for hand-correction, and again, the hand correction was completed in 8 hours.

This effort produced a combined set of 70 gold standard documents; the annotation cycles were complete and the learning cycles were begun.

A five-fold cross-validation test was run using an 80/20 split over the whole set of 70 documents. For each test, 56 documents were used for training and 14 documents were held out for testing. For each test, a different group of documents was held back for testing purposes.

For each cross-validation test, the method was the same:

- Run generic extraction only on the test set and save the output (i.e. extractions) to establish the baseline files
- Use the training set to run 3 different TBL learning cycles with the fnTBL toolkit
    - Using threshold 1
    - Using threshold 2
    - Using threshold 3

- Convert the rules learned by the fnTBL toolkit to format that can be used by the eQuery document processor in a transformation phase
- Apply the learned rules (transforms) to the test set and save the output (extractions) to facilitate evaluation
- Evaluate the learned rules by comparing for each test set the extractions from the baseline files, the transformed files, and the matching gold standard files for each threshold.

After running just one of the cross-validation tests, it became clear that there were some inconsistencies in the gold standard annotation between the first set of 35 documents and the second set of 35 documents. So the entire set of 70 documents with gold standard annotations was reviewed by the analysts again in an effort to remove the inconsistencies that had been present. This review and reconciliation effort took 16 hours of analyst time. When the review of the 70 documents was complete, and the gold standard annotations had been reconciled, all tests were re-run.

Evaluation and Results

The results of the tests were evaluated by comparing corresponding values from the baseline file, the gold standard file and the transformed file. The values that were compared were the types of events and the slots in the event frames of the files.

The results were evaluated using two complementary but slightly different methods. The first method used was to calculate desired changes, learned changes and correctly learned changes. Desired changes were determined by comparing the baseline extraction to the gold standard extraction; if these values differed, it was considered to be a desired change. Learned changes were determined by comparing the baseline extraction to the transformed extraction; if these values differed, it was considered to be a learned change. Correctly learned changes were determined by comparing the transformed extraction to both the baseline and gold standard extractions; if the transformed extraction was different from the baseline extraction and the same as the gold standard extraction, it was considered to be a correctly learned change. These three values were then used to calculate coverage and accuracy figures, which were in turn used to calculate an F-score:

Coverage (C) = Correctly Learned Changes / Desired Changes

Accuracy (A) = Correctly Learned Changes / Learned Changes

$$F = (2 * C * A) / (C + A)$$

The results obtained for the five cross-validation tests are shown below in Figure 5:

Test	**Threshold 1**	**Threshold 2**	**Threshold 3**

Test 1	Coverage = **68.29%**
	Accuracy = 88.42%
	F-score = **77.06%**
		Coverage = 64.50%
		Accuracy = **90.15%**
		F-score = 75.20%
			Coverage = 64.50%
			Accuracy = **90.15%**
			F-score = 75.20%

Test 2	Coverage = **54.05%**
	Accuracy = 88.21%
	F-score = **67.03%**
		Coverage = 50.33%
		Accuracy = 90.55%
		F-score = 64.70%
			Coverage = 48.36%
			Accuracy = **92.08%**
			F-score = 63.41%

Test 3	Coverage = **64.71%**
	Accuracy = 87.38%
	F-score = **74.36%**
		Coverage = 59.17%
		Accuracy = 86.80%
		F-score = 70.37%
			Coverage = 56.40%
			Accuracy = **88.59%**
			F-score = 68.92%

Test 4	Coverage = **54.47%**
	Accuracy = **91.16%**
	F-score = **68.19%**
		Coverage = 48.98%
		Accuracy = 90.60%
		F-score = 63.58%
			Coverage = 45.73%
			Accuracy = 91.09%
			F-score = 60.89%

Test 5	Coverage = **66.12%**
	Accuracy = 83.16%
	F-score = **73.67%**
		Coverage = 60.93%
		Accuracy = 85.77%
		F-score = 71.25%
			Coverage = 59.56%
			Accuracy = **87.55%**
			F-score = 70.89%

Figure 5: Five-fold cross-validation test results

In these results, as the threshold level increases, the coverage decreases and the accuracy increases. These figures for coverage, accuracy and F-score are comparable to what others have reported for similar tasks (Gildea & Palmer, 2002; Gildea & Hockenmaier, 2003).

The definitions that we used for coverage and accuracy were to compare the number of correctly learned changes in the text with the number of desired changes and the number of learned changes, respectively. However, this does not account for the situation where an incorrect change was learned from a change that was not supposed to be learned. So a more detailed set of scores was defined that increased the binary notion of “correctly learned” or “not correctly learned” to the ternary notion of “correctly learned”, “incorrectly learned” (also known as “errors”) and “not learned” (also known as “misses”).

The scores showed that the main problem with the training was with a high rate of “misses”. To examine this problem, frequency counts were computed for each event type or role label to be learned, and the number of training instances was counted. In one test set, for example, there was one value to be learned, the event type of the verb “kill”, that had 336 training instances. However, there were 90 learning items that had only one training instance. This reflects the fact that some verb constructs are sparse in the entire data, and without sufficient training data, transformations can’t be learned.

We speculated that the sparse data problem might affect only the less frequent event types and recomputed the scores using only the 6 “main” event types to get a coverage score of 67.93% and accuracy of 88.93%. But these scores were not significantly different from the other event types. An examination of the data shows that all event types, even the most frequent, have a few sparse verb constructs without enough training examples to learn from.

After the training and testing experiments were completed, a set of transformation rules was trained on all the annotated data. These rules, for the Russian Contract Killing domain, were delivered in August 2002.

TBL for domain specialization was used again in the project in the summer of 2003 to learn transformation rules for the event types and slotnames, representing relations, of the EE evaluation ontology.

III.B. Alias tracking, or Entity Resolution

Alias tracking is the ability of a system to recognize and unify variant references to a single entity. For this research area, within document alias tracking, which includes several types of name variance or coreference, was investigated. The major portion of this work was included in the delivery of December 2002, but some additional improvements were included in the final delivery of March 2004. In addition, a prototype for a system that could provide linguistic information for cross-document coreference was developed. This was demonstrated in the Entity Resolution module that was delivered in December 2002.

Alias Tracking within Documents

Within a text document, there may be several different forms of alias tracking, some of which are illustrated in this example text:

> *One early target of the* Federal Bureau of Investigation’s *Budapest office is expected to be* Semyon Y. Mogilevich*, a* Russian citizen *who has operated out of Budapest for a decade. Recently* he *has been linked to the growing money-laundering investigation in the United States involving the Bank of New York.* Mr. Mogilevich *is also the target of a separate money laundering and financial fraud investigation by the* F.B.I. *in Philadelphia, according to federal officials.*
> … The F.B.I. *will also have the final say over the hiring and firing of the 10 Hungarian agents who will work in the office, alongside five American agents.* The bureau *has long had agents posted in American embassies.*

In this example, there are four mentions of the following entity:
Federal Bureau of Investigation, F.B.I., F.B.I., the bureau.
and also four mentions of this entity:
Semyon Y. Mogilevich, Russian citizen, he, and Mr. Mogilevich

One form of alias tracking is sometimes known as name variance. This is the case with the use of the acronym, F.B.I. to refer to the Federal Bureau of Investigation, and the different forms of the names Semyon Y. Mogilevich and Mr. Mogilevich.

For the system, algorithms were developed for name variance, using patterns of names for persons, organizations, acronyms and some special cases of variance with upper case and punctuation. During entity detection, person names were analyzed for titles, and for first and last names and initials. These are used to recognize later mentions of the name. Mentions of organizations also have patterns where the main name occurs first, for example, "Unilever Corporation" could later be referred to as "Unilever". Acronym detection is done by examining the first one to three characters of each word of a name.

The other main form of alias tracking is that of coreference (sometimes the term coreference is used to include name variance as well). In the example text above, there is a pronominal coreference of "he" to Semyon Y. Mogilevich, and there is a nominal coreference of "the bureau" to the F.B.I. Both of these are examples of anaphora, where the referring phrase, "he" and "the bureau" occur later in the text than the phrase that they are referring to, known as the referent, in this case "Semyon Y. Mogilevich" and "F.B.I.". It is also possible to have cataphora, where the referring phrase occurs before the referent, not illustrated here.

Coreference resolution is a well-known problem that has been widely studied in the computational linguistics literature. It was implemented in the system as an algorithm to resolve nominal and pronominal anaphora. The basis of this algorithm is as follows:

1. First identify all potential referring expressions (definite noun phrases: e.g. the company, the officials, the killer; and pronouns: e.g. it, they, his)
2. Filter the potential referring expressions to remove those that are non-anaphoric.
    a. Remove existential uses of "it" and "there": e.g. "It is necessary", "There are none"
    b. Remove definite noun phrases that refer to general and not specific entities, e.g. "the world", "the galaxy"
    c. Remove other definite noun phrases:
        i. Time: "the sixties", "the past 16 years"
        ii. Geographical entities: "the Middle East", "the Atlantic", "the Nile"
3. Identify referent candidates (potential antecedents) for all references, essentially all noun phrases preceding the referring phrase
4. Compile and compare features (e.g. gender, number, animacy, type, recency, repeated mention, head match)
5. Score the candidates by the feature comparison and choose the best scoring candidate
    a. Weight assignment for scoring is similar to Lappin and Leass's Salience Factors (1994), modified to fit our feature selection and system parameters

Finally, an algorithm was developed for the resolution of cataphora, or forward coreference. Analysis of text showed that these occurred in a very limited form. The first form was that of an indefinite noun phrase, which could refer to a named entity either within the same sentence or the following sentence. This almost always occurred in the first paragraph of the article. In the following example, "a 35-year-old Soviet pop star" refers to "Igor Talkov".

> A 35-year-old Soviet pop star *was shot dead Sunday while giving a concert in St. Petersburg, the TASS news agency reported.* Igor Talkov *was shot through the heart at point blank range . . .*

The second form was that of a pronoun that could refer to a named entity within the same sentence. In this example, the pronoun "his" refers to "Mr. Berezovsky":

> *Speaking from* his *London office,* Mr. Berezovsky *said that he had spoken to* . . .

For these types of cataphora, referring phrases were identified and the resolution was to the named entity of the same type in that or the next sentence.

Linguistic Features for Cross Document Alias Tracking

Cross document alias tracking is the task of deciding whether names from different documents refer to the same entity or not. This problem has two aspects: two different names may refer to the same entity, where these may be different forms, different spellings, or intentional aliases of the same entity; or occurrences of the same name in two different documents may refer to different entities, for example, there may be more than one person named "Michael Jordan".

Many aspects of such alias tracking algorithms lie outside the scope of this project, since they may involve knowledge of specific languages for name spelling variations, to detect when two names refer to the same entity, or they may involve knowledge sources to find identifying features of entities, such as address or phone number, to detect when one name refers to two different entities. However, the processing of documents by an NLP system can produce information about entities other than just the name as it occurs in the document. In this part of the project, a prototype system was implemented that demonstrates how such information can be used in the cross document alias tracking problem.

A prototype cross-document Entity Resolution system was designed in order to investigate and demonstrate how such a system could use linguistic features, stored in extractions, from an NLP document processing system in Alias Tracking. Forefront in this capability were the goals of named entity resolution in a multiple document, multiple source, multiple genre environment in which new documents continue to arrive and to be processed and added to the collection of document extractions with hypothesized name links. In particular, this algorithm never assumes that the document collection is closed, but always adds the names to the current state of the document collection. It is expected that additional name equivalences can be discovered by LD groups using structured data sources. For purposes of demonstration, the entity resolution capability focused on the problem of deciding when two different names in different documents actually refer to the same entity.

In deciding which names in different documents are to be equivalent, a combination of features used in intra-document coreference was considered, primarily gender and animacy, and additional features that are directly extracted from the document. In the latter class of features, type, title, firstname, lastname, aka and acronym fields are used, and this can easily be extended to other fields, such as date-of-birth and address. But recall that these fields are only used for information that is directly stated in the document and this information is only intended as a supplement to structured information that can be obtained from other sources.

In order to make a more interesting demonstration, a small version of an alternate name spelling algorithm for Russian was implemented. This came about because the documents were from the Russian Contract Killing dataset and a native Ukrainian speaker was available and working on the project who could easily implement such a Russian name transliteration scheme. This scheme recognized English character classes that came from original Russian characters and allowed these as alternate spellings.

This algorithm was tested on a collection of 609 documents from the Russian Contract Killing corpus, on all the types of entities which can be persons, organizations or locations. It was correctly able to demonstrate the equivalence of 146 entities with different names across the document collection. Examples included cases such as

Boris Abramovich Berezovskiy and Boris Berezovsky

which used a combination of firstname, lastname and Russian alternate name spelling rules.

One particularly interesting example was where several documents referred to a person also known as “Arkan”. But one document included the phrase “Zeljko Raznjatovic, also known as Arkan”. This document included an attribute “aka” for the entity Zeljko Raznjatovic, from which the entity resolution algorithm could conclude that these were the same entity.

The Entity Resolution module was delivered in December 2002. The single document coreference system was also delivered at that time, and an improved version was included in the final delivery of March 2004.

III.c. Temporal Sequencing and Extraction for Scenarios

An important aspect of the understanding of the relationships between the entities and events in natural language text is the temporal aspect. Associating time with events and understanding the temporal sequencing between events, gives the important event relation building blocks that can be used to understand how the events fit into more complex scenarios. In this project, research was carried out in both of these aspects of temporal extraction; this section provides a combined report. Parts of this section are taken from the paper (Symonenko, McCracken and Liddy 2004).

The problem of extracting temporal relations can be analyzed into the following parts. First, there is the task of recognizing temporal expressions, that is, expressions such as “last Monday” and “two decades” as well as explicit dates. Then there is the task of recognizing temporal relations that anchor events to a temporal expression; these can be characterized as event-time relations. Finally, there is the task of recognizing temporal relations that describe the sequencing between events; these can be characterized as event-event relations.

Considerable research has been carried out in the Artificial Intelligence domain about models of temporal concepts and their representation, and while the NLP community has long recognized the importance of temporal concepts in systems such as James Allen’s temporal algebra (Allen, 1983), there has not yet been any automatic extraction of temporal relations in English natural language text that extends beyond a few relations.

The TIMEX project, part of the DARPA TIDES program started in 1999, focused on the development of guidelines for annotating temporal expressions and annotating a corpus to be utilized by developers of systems[2] (Gerber et al., 2002; Ferro et al., 2004). A related task is the work on creating the markup language, TimeML, to represent temporal expressions, and developing the TempEx system for automatic annotation and normalization (i.e. converting to the ISO format) of temporal expressions (Wilson & Mani, 2000). It is important to note that TimeML and DAML temporal representations are compatible in both temporal units and temporal relations (Hobbs, Pustejovsky, 2002).

Automatic recognition of temporal relations has been a focus of a number of recent studies. Filatova & Hovy (2001) applied an event-anchoring system to arrange distinct news stories about the same event on a timeline by associating an event-clause to an explicit temporal reference. Evaluation on a set of 6 news stories about an earthquake showed a performance of 52% compared to human judgments. Schilder & Habel (2001) describe a rule-based system of relating temporal expressions to events in the news texts in German. Their model of temporal relations follows Allen’s temporal algebra. Semantic models for anchoring made use of the temporal semantics of prepositions (such as *at*, *by*). The system was evaluated on a small (10 articles from *Financial Times*) corpus and was reported to achieve above 90% precision and recall in tagging and about 85% in event anchoring. Mani et al. (2003) experimented with machine learning, using a C5.0 classifier, applied to the task of identifying temporal relations, and reported 84.6% accuracy in temporal anchoring and 75.4% in partial ordering of events.

This project’s work on temporal relations followed the temporal modeling work of these previous efforts and extended the capability of the numbers and type of temporal relations that can be automatically extracted from text. The work was first developed on the Russian Contract Killing corpus, and was further developed and tested on the dataset used for the 2003 EELD evaluation. This corpus was compiled by Global Infotek (GITI) from US, British, Greek, Romanian, Russian, and Ukrainian print media sources. Both corpora included short news and longer news stories (e.g., stories from the EELD training set vary in size from 2 to 41 KB).

In the news genre, the story line is different from a “narrative convention”, where events are presented in chronological order. In particular, the temporal structure of news is guided by the perceived news value rather than chronology: the major news line is presented first, and usually relates to the most recent event. It is worth adding that, from

[2] In March 2004, the annotated TimeBank Corpus (news documents) was released: http://nrrc.mitre.org/NRRC/Docs_Data/MPQA_04/approval_time.htm

our observations, the more detailed account following the headline still typically adheres to the "narrative convention".

Our system uses the temporal relation categories from Allen's (1984) temporal logic, as well as the TimeML event-based model of temporal aspects (Advanced Research and Development Activity (ARDA) Workshop) as the basis of its ability to associate a temporal expression with an event, or to assign a temporal relation between events and/or entities. Some changes in the relations were made to fit the corpus and to better label the concept for the user. The following table outlines the temporal relation types that were developed for the EELD project. (Note that the original development of these temporal aspects followed a more complex scheme adapted from TimeML, and was in the delivery of May 2003. In our later development, the temporal aspects were simplified into these more "relation-like" concepts, and form the basis both of this report and the final March 2004 delivery.)

Aspect	**Explanation**	**Example**
Occurs	an event happens at a certain point in time	*Abdul Radzhabov, General Director of the Daginterstroj building firm, was **killed** at point-blank range in the city of Kaspiisk at **19:15 on Wednesday***
Holds	an event takes place over a period of time	*Nikolay Bykalov, who **worked** as Lyubarskiy's personal driver **for 10 years**.*
Since	an event takes place at some time before/after (and also the same time as) a certain point in time or another event	*The annual quota has risen each year **since 1996**, but stocks have been on a downward slide **since 1982**, the group said.*
Frequency	an event happens a number of times, at regular or irregular intervals	*The WWF concluded after a year-long investigation that Alaskan pollack was at immediate risk, as the amount **fished** from the Bering Sea **each year** exceeds the quota by an estimated 150 percent.*
Date-of-birth/death	a special event case, which cannot be adequately represented by any of the above temporal aspects. It, obviously, is an entity attribute only.	*Romuli Kikaleyshvili, **born in 1962**, Georgian, leader of the Georgian group..*
Before/After	an event takes place before/after (but NOT at the same time as) a certain point in time or another event	*Odeh was **arrested** in Pakistan **after arriving** on a flight from Nairobi* *As a matter of fact, **after 1990**, the Valeologia Company has been quite often **charged** with armament and radioactive substances smuggling[144.sgm,S72].*

Concurrent	an event takes place at the same time as another event	*Last Wednesday morning, two men* ***walked*** *up to Novosyolov's car as it was* ***stopped*** *at a traffic light*
Included_In	an event takes place at some time within the period of the more lasting event	

Figure 6. Temporal aspects employed by the CNLP temporal extraction system

In order to implement the automatic extraction of these relations, several phases were added to our extraction system.   These phases identified temporal expressions, categorized temporal expressions as time, date and interval categories, and finally, extracted temporal relations.

The temporal relation extractions were implemented as patterns involving semantic clues in a set of rules. The semantic clues currently employed include prepositions, conjunctions, adverbials, certain verb groups, and particular phrases. For example, such prepositions as *at*, *of*, *around,* when followed by a temporal expression, serve as a clue for the temporal event attribute *occurs*; and prepositions *for, throughout, within, during* indicate the temporal event attribute *holds*. Certain verbs, such as *inhabit, live, last, reside, spend,* and others are also used as clues for the *holds* temporal relation.

The heavyweight man has ***managed to establish*** *a multinational firm from scratch* ***within five years*** *with 5,000 employees ...*

event = establish
object = firm
agent = man
holds = five years
extent = managed to establish

In the eQuery frame representation, the temporal slots include both event anchoring relations and event sequencing relations. The distinction lies in a slot value. Whereas event anchoring slots (*occurs, holds, since, frequency, date-of-birth/death*) take temporal expressions as values, event sequencing slots (*concurrent, included_in*) take other events as values. *Before/after* slots are of a dual nature, as they can take both time expressions and events. Observations show that the majority of the *before/after* cases in the training and evaluation sets indicated sequencing between events.

Odeh was ***arrested*** *in Pakistan* ***after arriving*** *on a flight from Nairobi* ***on August 7****, the day of the* ***bombings****.*

event = arrested
after = arriving
object = Odeh
location = Pakistan

event = arriving

occurs = August 7
concurrent = bombings

An event/entity can also take a *temp_qual* slot, which further specifies the modality and/or negation aspects for the event. Its value is taken directly from the context that the event/entity occurs in. In the following example the *temp_qual* slot communicates important information about the likelihood of the event occurrence. Modality may also include that the event is unlikely to happen or may happen in the future.

Maslov responded by calling Makarenko a "semi-criminal businessman with no place in politics" and suggested that backer of the former governor ***may have tried to execute*** *Makarenko.*

event = execute
temp_qual = may have
extent = tried to execute

The *temp_interval* slot is used when, along with the temporal sequencing relation between the two events, the temporal interval between them is also specified in text.

A few days after his victory in court, Mazurin ***died*** *in a car accident . . .*

event = died
after = his victory
temp_interval = a few days
location = car
agent = Mazurin

The system was evaluated on the subset of the EELD corpus that was used for the overall EELD evaluation in August 2003. First the recognition of temporal expressions was evaluated: the system demonstrated an effective performance on these, achieving 99.1% in F-measure.

Next the temporal relations were evaluated; system output was analyzed by a CNLP research analyst who judged relations to be *correct, false, missed* or *wrong,* according to the CNLP model of temporal relations. Precision and recall were calculated following the standard procedures. Partial (half-) credit was given to the relations identified correctly, but which were anchored to the wrong event/entity. The F-measure was calculated following (Van Rijsbergen, 1979), with b=2, which gives equal weight to recall and precision.

Temporal Relation	**Precision**	**Recall**	**F-measure**
Occurs	70.5	72.0	71.3
Holds	63.6	45.2	52.8
Since	64.3	61.4	62.8
Frequency	60.0	75.0	66.7
Before	73.3	55.0	62.9
After	53.1	84.3	65.2
Concurrent	51.1	57.5	54.1

Date-of-birth	100	100	100
Date-of-death	80	100	88.9
Overall	**68.5**	**72.3**	**69.4**

Figure 7. Temporal relation extraction: performance on the EELD evaluation dataset.

On this dataset, we also measured the frequency of the relations that were evaluated. The relation *occurs* constituted a substantial majority – two thirds of all those identified.

Temporal Relation	Frequency of Occurence
Occurs	65%
Holds	10%
Before	3%
After	8%
Since	3%
Concurrent	6%
Frequency	2%
Date-of-Birth	2%
Date-of-Death	1%

Figure 8. Frequency of temporal relations in EELD evaluation dataset.

Results reported herein favorably compare to similar research on automatic extraction of temporal relations, including event anchoring and event sequencing. As can be seen from Figure 9 and mentioned above, the system developed at CNLP performs well at tagging temporal expressions, with precision of 99.5% and recall of 98.7%. The system performance numbers for event anchoring and event sequencing are less impressive per se, but are still a good result given the diverse genre and the larger number of relation types. The main goal of capturing significantly more types of temporal relations (namely, 11 as compared to 1 to 3) was achieved.

Reported in	**Relations, #**	**Texts, #**	**Genre**	**Tagging**		**Event Anchoring**		**Event Sequencing**	
				Prec.	***Recall***	***Prec.***	***Recall***	***Prec.***	***Recall***
Mani & Wilson	1	221	single	83.7	82.7	59.4			
Mani et al.	3	6	single			59.0		73.7	77.7
Filatova & Hovy	1	6	single			82.0			
CNLP	11	40	diverse	99.5	98.7	73.4	72.6	58.5	65.8

Figure 9. Comparison of annotation and anchoring of temporal expressions across related research

III.d Linguistic Inferencing for Event Coreference

CNLP's extraction system is based on a frame representation of the information from text, where the contents of the frame slots are natural language phrases. There are several applications for which this frame information can be further processed to satisfy a specific goal. Goals may either find particular pieces of information, such as for building profiles of entities, or they may be to find relations between entities and events in the frames. The frames form the basis of a natural language frame logic, which has a type of inferencing referred to here as linguistic inferencing, to find instances of goal frames. The inferencing is linguistic because the set of rules it is based on, the axioms of the logic, use linguistic information either from the natural language phrases themselves or from the form of the frame representing the text. This linguistic inferencing system is used to solve the event coreference problem.

The event coreference problem is that a text may give two different mentions of the same event. For example, in the following text, a human reader would conclude that the two highlighted verb phrases were referring to the same event.

A 35-year-old Soviet pop star ***was killed*** *Sunday while giving a concert in St. Petersburg . . .*
Igor Talkov ***was shot*** *through the heart at point blank range by an unidentified spectator . . .*

From the system's extraction viewpoint, the two verb phrases would give two extracted event frames, and the event coreference algorithm must decide if they are, in fact, two mentions of the same event. The first sentence will produce a frame such as:

```
event = kill
type = Murder
  victim = Soviet pop star
  occurs = Sunday
  concurrent = give concert
  eventOccursAt = St. Petersburg
```

The second sentence gives a frame which has the same event type and in which the victim role has values which are equivalent under entity coreference.

```
event = shot through the heart
type = Murder
  victim = Igor Talkov
  perpetrator = unidentified spectator
```

In the implemented frame logic, the inference rule sees one of these frames as a goal and tries to show that goal by showing that each of the values of matching slots can also be shown to be equivalent as sub-goals. The inference rule is also abductive (Hobbs 1993), which gives a probability of matching the goal frames based on the probabilities of matching the sub-goals. This form of abductive inference was found to be appropriate for the event coreference problem because some slots for an event type should be given higher probabilities to be matched, and because this type of inference allows successful matching even when some information is missing, which is the normal case in extracted events, i.e. not all event semantic roles are given for each mention of the event.

Different types of events in text were analyzed and a system of weights was created to give higher weight to important slots and lower weight to slots that are less important. For example, in events of type Murder and AttackOnTangible, the “performedBy” and “victim” relations are important in an event mention. In events of type ArrivingAtAPlace and LeavingAPlace, the “eventOccursAt” relation is important. These weights are used by the inference engine to weight the importance of the slots in determining a match.

During the analysis of event mentions in text, it was observed that many of the event mentions were nominalizations (nouns), called nominal references, to the event. However, some of these nominal references also had roles, such as in the phrases:

> *Tkachuk’s death*
> *the death of Tkachuk*

The system extracted these events with roles and the inferencing technique was used for event coreferencing.

However, it was observed that other nominal event coreferences did not have roles and were not appropriate for the inferencing technique. For example, in the following passage, the event mention of “detained” is followed by another mention of the same event as “the current arrests”, where no roles are given.

*Around one hundred people have already been **detained** in Europe during the Spider Web operation to combat the "Russian mafia." Italian Internal Affairs Minister Claudio Scajola is promising that the current **arrests** will be followed by other scandalous unmaskings.*

These references primarily occurred with the definite determiner “the” and typically occurred within four sentences after the event was mentioned as a verb, and it was assumed that entity coreference could be applied to these cases.

The inferencing-based event coreference algorithm was included in the September 9 delivery, and also in the final delivery of March 2004.

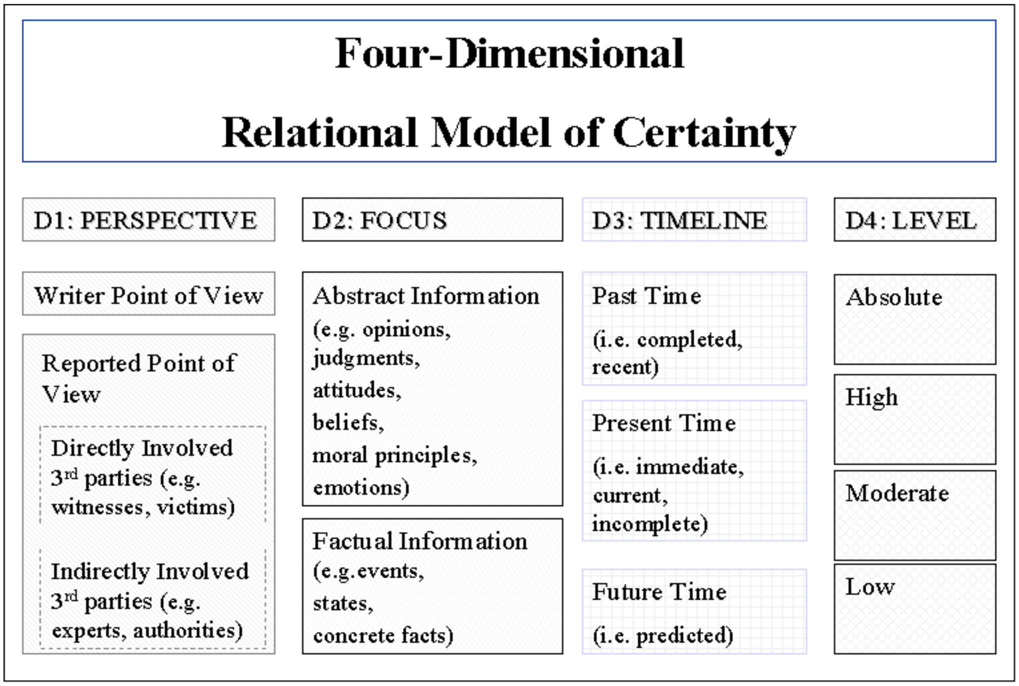
III.e Confidence Levels using Linguistic Certainty

This section provides an overview of the certainty analysis research and the development of an experimental model, preliminary testing of inter-coder agreement, and an evaluation of system performance results of certainty detection and categorization.

Theoretical Background

An innovative experimental module has been implemented for automated certainty detection and categorization based on theoretical work presented in Rubin, Kando, and Liddy (2004) that is summarized and exemplified below. (For more details please see the paper by Rubin, Kando, and Liddy (2004).)

Certainty is typically defined as "*the quality or state of mind of being free from doubt, especially on the basis of evidence*" (Merriam-Webster 2003). Rubin, Kando, and Liddy (2004) suggested a theoretical framework that extends the definition in the context of news article analysis by distinguishing 4 relational dimensions used in analysis and categorizations of the types of certainty. The dimensions include *Perspective, Focus, Timeline* and *Level* of certainty, depicted in Figure 10 and described below in more detail.



Four-Dimensional
Relational Model of Certainty
D1: PERSPECTIVE
D2: FOCUS
D3: TIMELINE
D4: LEVEL
Writer Point of View
Reported Point of View
Directly Involved 3rd parties (e.g. witnesses, victims)
Indirectly Involved 3rd parties (e.g. experts, authorities)
Abstract Information (e.g. opinions, judgments, attitudes, beliefs, moral principles, emotions)
Factual Information (e.g.events, states, concrete facts)
Past Time (i.e. completed, recent)
Present Time (i.e. immediate, current, incomplete)
Future Time (i.e. predicted)
Absolute
High
Moderate
Low

Figure 10. Hypothesized dimensions of certainty. Reprinted from Rubin, Kando, and Liddy (2004).

Perspective is the point of view or voice of the experiencer of certainty. There are two major categories – *the writer's perspective* and *the reported point of view* which can be further sub-divided into the third parties that are *directly* involved in the events, and the ones that are *indirectly* involved. The writer is the author of the article. For example, the writer's certainty is reflected in the following statement:

*"Dead men, **of course[3],** do not talk, but three of the shooting victims managed to survive."*

[3] The bolded areas in examples correspond to certainty markers that explicitly identify certainty

The third parties are people or organizations that are directly involved (such as victims, participants, survivors) or indirectly involved in some professional capacity (such as experts and authorities, other reporters), as exemplified below:

"They are in the hospital and, ***according to the prosecutor****, have* ***a good chance*** *of regaining their health. "*

The second dimension of *Focus* distinguishes abstract from factual information pertaining to expressed certainty. The abstract focus of certainty is an idea that does not represent an external reality but rather a hypothesized world, existing only in the mind, separated from embodiment or object of nature, such as emotions, opinions, judgments, attitudes, beliefs, moral principles. Consider this opinion that necessitates an action:

"But while doing so, he [Richard Holbrooke, the United States' permanent representative] ***must also work*** *on trouble spots like Iraq, Kosovo and East Timor where timely U.N.* ***action is imperative****."*

Factual information is based on, characterized by, or contains facts, i.e. has actual existence in the world of events, such as events, states, concrete facts. For instance:

"Many outside experts ***wonder if*** *the Taliban* ***actually*** *helped the hijackers escape,* ***perhaps*** *over the nearby border to Pakistan or into the hills of southern Afghanistan where Islamic terrorist training camps* ***are believed to operate***

The third dimension of *Timeline* simply reflects relevance of time to the point of reference, the writing or publication of the news report. This dimension records whether the event about which the certainty is expressed already took place (past), or whether it is a current state of affairs, or alternatively whether it is a prediction of future events (future). An example of certainty in the past is given below:

The failure lasted only about 30 minutes and had no operational effect, the FAA said, adding that ***it was not even clear*** *that the problem was caused by the date change.*

The most important distinction is drawn in the fourth dimension. Most interesting is to automatically detect what *Level* of certainty the person expresses about the events in focus in a given timeframe. Certainty falls naturally into several levels. The original model in Rubin, Kando and Liddy (2004) suggests a four way distinction – *absolute, high, moderate,* or *low* in the statements that have explicitly marked certainty information. Here are a few examples:

Eventually, however, *auditors* ***will almost certainly have to*** *form a tough self-regulatory body that can oversee its members' actions…* <<ABSOLUTE>>

… but ***clearly*** *an opportunity is at hand for the rest of the world to pressure both sides to devise a lasting peace based on democratic values and respect for human rights.* <<HIGH>>

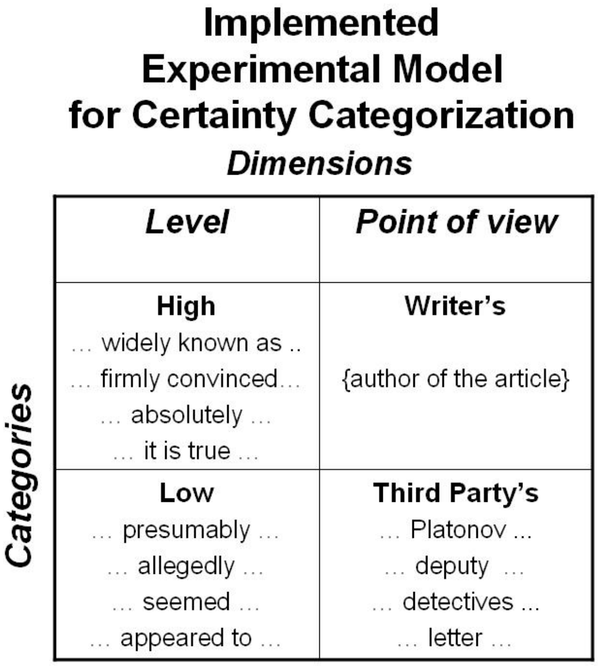
That fear ***now seems*** *exaggerated, but it was not entirely fanciful.* <<MODERATE>>

information in a given statement.

So far the presidential candidates are more interested in talking about what a surplus ***might buy*** *than in the painful choices that lie ahead* <<LOW>>

Experimental Model

The experimental model implemented for the EELD project incorporates two extreme levels of certainty – *high* and *low*, and two major perspective categories – *writer's* and *reported point of view* – in order to test the feasibility of the automated detection and categorization of certainty.



Implemented Experimental Model for Certainty Categorization

Dimensions

	Level	*Point of view*
Categories	**High** … widely known as .. … firmly convinced… … absolutely … … it is true …	**Writer's** {author of the article}
	Low … presumably … … allegedly … … seemed … … appeared to …	**Third Party's** … Platonov ... … deputy … … detectives ... … letter …

Figure 11. Currently Implemented Experimental Model

High Level of certainty is an explicitly assertive statement that sounds more confident than normal. *Low Level* of certainty is an explicitly uncertain statement that sounds unsure. Writer's own perspective such as

*"**It seems** that **no one any longer doubts** that this market is criminal today*

is contrasted to the reported certainty expressed in statements by other people:

I will quote ***an excerpt*** *from it, preserving the original style and spelling: "**According to available results**, there are **reasons to believe** that this arrest is aimed at achieving visible results but does not have any legitimate grounds."*

Applications

Possible applications include alerting analysts to the level above or below normal and associating certainty with its source; searching by level and the point of view parameter;

summarizing by document, across-documents, topic; inferring importance, popularity of opinion, true state of affairs from high level certainty statements from multiple sources.

In the alerts application we flag extreme cases of *high certainty* or *low certainty* for intelligence analysts and associate the extremes with their source (points of view: writer or other). Then analysts can interpret what is the likely reason for those extremes depending on the source and they can make further judgments using their world knowledge. A cardiogram analogy was used to represent a flow of narrative in news (Figure 12).

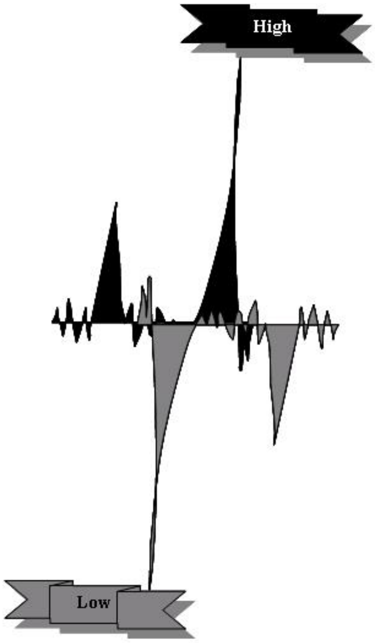


Figure 12. Alert System Analogy. Texts have varying levels of certainty with its extreme highs and lows that are flagged by our certainty analysis module.

The flow of narrative in texts contains different levels of certainty, frequently unnoticed. The flags mark the extremes of high and low certainty about an alert.

The envisioned searching applications will provide the ability to search a multitude of texts by one of the certainty parameters (source, focus, timeframe, and level) and sort retrieved information accordingly. It would allow the user to decrease the amount of uncertain information, prioritize sources that provide highly certain information, and identify sources with absolute judgments that can become suspect for being wrong. It would also provide the ability to ask better overview questions about a particular source, for instance, “What does President Bush sound most certain about in his speeches, and what is he uncertain about?”, a focus of certainty or “Which aspects of the Middle East crisis do people exhibit least certainty about?”, or highlight particularly extreme case of absolute certainty “What did President Clinton emphasize with absolute certainty?”.

Data

Four sets of data from the Russian Contract Killing (RCK EELD) files were used. Training data included 9 RCK files (489 sentences). Refinements of the guidelines and patterns for manual certainty marking were performed on 10 – 15 additional RCK files. The inter-coder agreement study was accomplished on 5 RCK files (431 sentences). The gold standard used to evaluate the system performance contained a subset of agreed upon clues from the sentences in the 5 RCK files (113 instances of clues in sentences).

Module Development and Implementation

A set of certainty markers (i.e. textual clues) was identified for either high or low levels, e.g. *must, certainly, absolutely, for sure* or *may, might, unclear, remains to be seen,* and a set of guidelines was developed based on manual analysis of RCK data. The system module that identifies clues and patterns and categorizes them as high or low certainty with writer's or others' points of view was implemented, and a small inter-coder agreement study with 2 coders was conducted. An Excel annotation tool for data collection, and a calculation tool were developed. A gold standard was created and the system was run against the gold standard data to evaluate the system's performance.

As a result of the certainty analysis and extraction, each text may contain one or more certainty extractions containing *a certainty frame* attached to a particular sentence. Each certainty extraction has

ctclue – the certainty clue is the text in the sentence giving rise to the certainty;
level – either high or low if this certainty is above or below normal;
ptview – the source of the certainty.

Here are a few examples of the resulting extraction frames:

The only thing that is ***known for certain*** *is that the real perpetrators responsible for the collapse of the Black Sea Shipping Company have not been named to this day, and the death rate among those in the know about ChMP funds is high.*

ctclue = for certain
level = high
ptview = writer

However, ***local journalists insist*** *that the source of the earlier information about the charge being "driving someone to suicide" had been the oblast prosecutor's office itself.*

ctclue = insist
level = high
ptview = local journalist

It is still ***unclear*** *which theory is correct…*

ctclue = unclear
level = low
ptview = writer

Evaluation: Inter-coder Agreement

An inter-coder test was run using 2 coders (we will call them S and V), to determine if they could identify and agree on the presence of clues in texts, and it was found that the two of them together identified 125 clues. They "blindly" agreed on 67 (53.6%). Several instances (12) were overlooked by coders (S missed 5 and V missed 7) but were then added in independently by looking at each other's results before discussing them. Thus, the adjusted agreement rate was **63.2 % (79 out of 125)**. After a discussion with the purpose of creating a gold standard, the coders agreed to use **113 clues** out of **125 (90.4%)** as the gold standard.

Once the clues were identified the raters agreed on *high level* of certainty - 95 times, and on *low level* of certainty - 17 times. Cohen's Kappa is a measure that takes into account agreement that could occur by chance. Cohen's Kappa statistic for level agreement was .94 which is considered extremely high[4].

Out of the 113 clues, 7 had a slightly different extent in terms of the number of words included, for example, S chose *convinced* and V marked *firmly convinced*. There were 104 cases of writer's point of view and 8 cases where the points of view were other than the writer's (out of 113 clues). There was one case of point of view disagreement between the coders.

Evaluation: System Performance Results

The adjusted set with a high complete two inter-rater agreement (113 clues, 90.4% agreement) served as the gold standard for the system performance. Out of 113 gold

[4] **Cohen's Kappa Value Interpretation Scale**

According to Landis & Koch (1977) here is how the kappa rates should be interpreted:

Below 0.00	Poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost perfect

Gradner (1995) recommends that the value exceeds .70 before you proceed with further data analysis (cited in http://www.utexas.edu/cc/faqs/stat/general/gen27.htm (Accessed 02/24/04) standard clues, **82 (72.6%)** were identified by the system correctly. The system also had a false positive rate of 5% (or 6 clues).

Out of 82 instances of correct clue identification by the system, **67 (81.7%)** were correctly identified as the high level of certainty and 12 (14.6%) as the low one, the error rate was 2.5%. The Kappa statistic for the system and gold standard agreement on level was .88 which is considered very high.

There were also **74 correct matches (90.2%)** for the writer's point of view (out of 82 instances of correct clue identification). There was insufficient data to reliably assess the accuracy of the reported point of view due to its rarity of occurrence in texts. Not enough instances of alternative points of view with certainty expressions were present in the selected data files and, consequently, in the gold standard. Those are rare occasions and are not easily collected from a relatively small subset of narratives. It will require further data collection, training, and testing.

Conclusions and Future Work

Overall, the development of the experimental model, the preliminary evaluation of the inter-coder agreement, and the system performance results have demonstrated the feasibility of certainty analysis and extraction and categorization with reasonable system performance results.

The system performed reasonably well against the gold standard:

- 73% of clues were correctly identified
- Cohen’s Kappa = .88 for level agreement
- 90% of writer’s points of view were correctly identified

The main challenge for certainty analysis is recognizing and identifying certainty clues. Even manually, two coders had some difficulties:

- 54% “blind” agreement (before discussion)
- 63% agreement adjusted for omitted cases (before discussion)
- 90% agreement after discussion (used for gold standard)

Once the clues were isolated manually, the agreement between the two coders on the level was high

- Cohen’s Kappa = .94

Future work will include

- Improving the existing experimental model
- Expanding the inventory of extracted clues and patterns
- Expanding the experimental model to the original full size – more levels of certainty in all four dimensions as suggested in Rubin, Kando, and Liddy (2004).

Other possible improvements and application development may include extracting sets of certainties per author or entity in texts, reconstructing people’s beliefs and comfort level about topics, identifying strength and weaknesses (in terms of certainty and relying on own opinions of others', identifying distributions of certainty types per text genre and summarizing certainties within and across texts, as well as implementing visualization of relations of entities, foci and levels.

IV. Evaluation and Generic Extraction Development

In addition to the main research areas described above, this project continued the development of generic extractions, which were initiated under the EELD Seedling Project Funding.   This consisted primarily of the entity relation extraction, the further development and extraction of relations between events and entities. The latter relations are based on the semantic roles of the generic events. These improvements to the generic extraction system were evaluated firstly by an internal evaluation based on the CNLP taxonomy.   It was also evaluated by the program-wide external evaluations conducted as the ACE evaluation in August 2002 and the EELD evaluation in August 2003, which was based on the EE subset of the EELD ontology.

The internal evaluation was conducted to measure directly how well the rule-based system identifies concepts, categorizes concepts and extracts entity and event relations, based on the mentions of entities in the text. The internal evaluation was carried out by CNLP linguistic analysts by annotating gold standard text and comparing it with the system output.

The evaluation of the EELD Seedling Project was used to help set goals for the EELD project. The Interim Project evaluation was conducted on the nuclear smuggling corpus. The evaluation reported on identification and categorization of named entities and numeric concepts and on the initial entity and event attribute extractions that were the basis for the initial generic extraction rule set developed under that project. Based on those results, shown in Figure 13, goals for the EELD project were set.

	Interim Base		Interim Final		EELD Goal	
	precision	recall	precision	recall	precision	recall
Named Entity Identification	92%	92%	92%	94%	95%	95%
Named Entity Categorization	97%	77%	97%	71%	97%	85%
Numeric Concept Identification	81%	82%	93%	98%	95%	95%
Numeric Concept Categorization	90%	90%	94%	94%	90%	65%
Entity Attribute (Relation) Extraction	71%	13%	87%	44%	90%	65%
Event Attribute (Relation) Extraction	64%	22%	69%	48%	85%	65%

Figure 13 Evaluation results from the EELD Seedling Project and goals for the EELD project

For the EELD internal evaluation, the evaluation corpus was run on the Russian Contract Killing (RCK) corpus. The Gold standard consisted of 31 documents randomly chosen from this corpus. A team of 4 CNLP linguistic analysts analyzed and performed an initial annotation of these documents, based on the CNLP generic extraction model. Then the analysts reviewed the combined corpus and discussed and revised the annotation.
The CNLP analysts then performed an initial baseline evaluation for this corpus, using the CNLP eQuery extraction module of September 26, 2001, which was effectively the system at the start of the EELD project.

The next evaluation was the external ACE evaluation, conducted in August 2002. Since this evaluation used the ACE taxonomy of entity and relation types, the CNLP extraction development during the spring and summer of 2002 concentrated on ACE-style extraction. For this, the CNLP concept identification was essentially kept, as it was similar, although not the same, as the ACE concept of entity mention “head”. The rules of CNLP named entity categorization were converted to the ACE entity types, and new rules were developed for common noun, or nominal, entity categorization. Finally, a new set of entity relation rules was written for the ACE entity relations. This revision of the CNLP system to ACE was not complete, as the training data was received on June 24, 2002, and the evaluation was conducted on August 19, 2002. Essentially, a parallel system was developed and not completed in the time available.

In March of 2003, the CNLP analysts conducted an internal evaluation, based on the extraction development carried out in the fall of 2002 and early 2003. This development included the adaptation of nominal entity categorization and ACE relation rules back to the CNLP system. Further development of rules to extend the coverage of generic extraction to additional verb constructs and also to nominalizations was included in a steady increase in extraction coverage.

The results of this evaluation are reported in the following charts. For each of recall, precision and F-measure, the EELD baseline score and the March 2003 score, which are labeled as “Post-ACE”, are reported. These two scores are also compared with the EELD goals that were established at the beginning of the program.

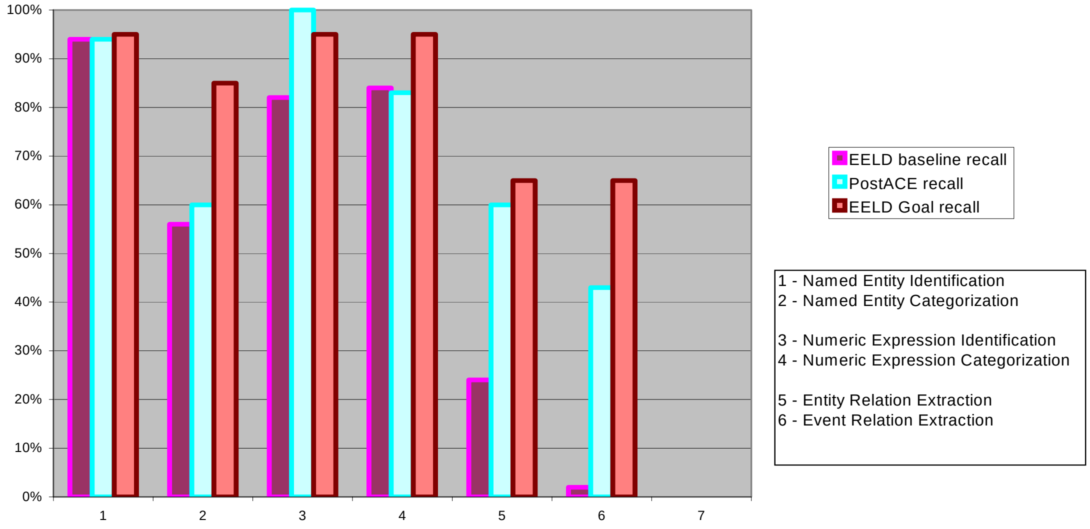


Figure 14 EELD project results: Recall

Figure 14 shows a 25% improvement in the identification of numeric concepts, while retaining the same precision in categorizing these concepts. It also shows that more than twice the number of entity relations are extracted, and a twentyfold increase in the number of event relations that are extracted.

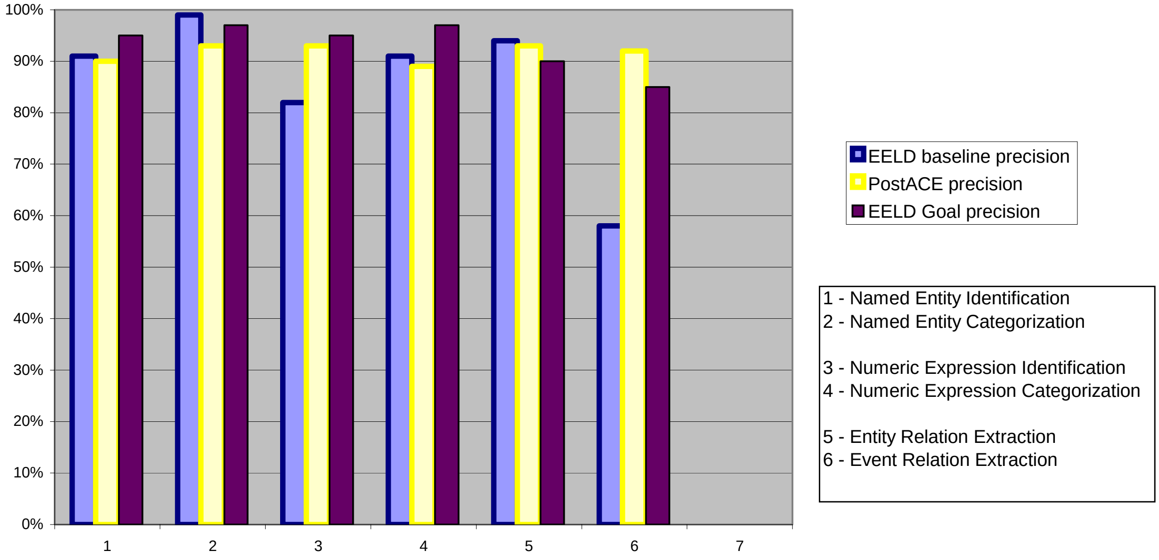


Figure 15 EELD project results: Precision

Figure 15 demonstrates an improvement in the precision of numeric concept identification and in the precision of event relations that are extracted (by one third). The improvements are significant because for both of these types of extractions, recall also improved significantly. Often, within the information field, an increase in recall negatively affects precision. In this case, both have improved.

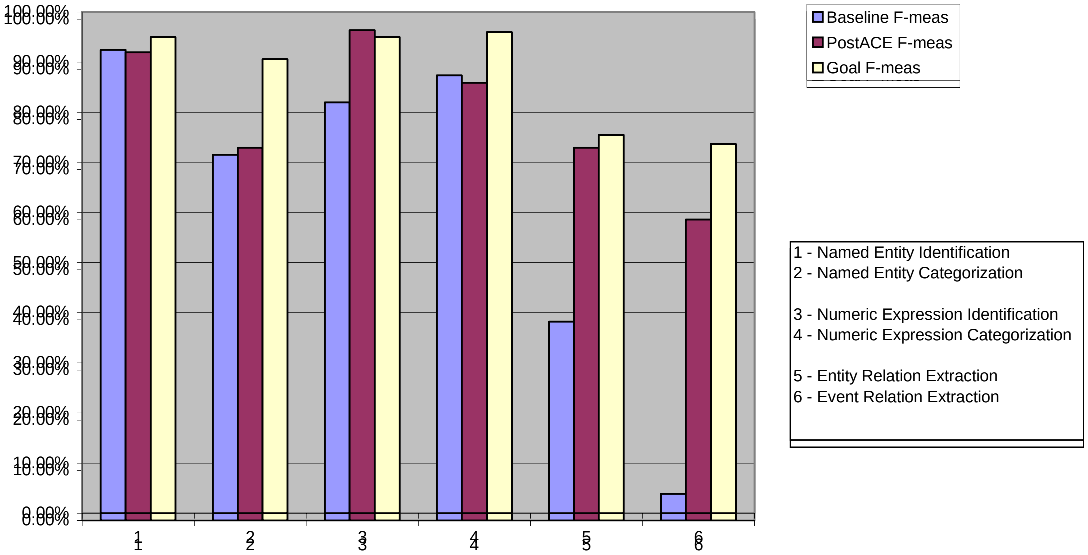


Figure 16 EELD project results: F-measure

The F-measure is a way of showing the overall increase/decrease in recall and precision together. There is little change in identification and categorization of Named entities, which is to be expected since this was not the focus of the EELD work, and they were already very high. The graph shows that identification of numeric concepts improved by about 25% without losing categorization capability.

Most significant, and the main focus of EELD work, is the dramatic improvement of relation extraction. Entity relation extraction improved by a factor of 2, and event relation extraction improved by a factor of 15.

The final evaluation for the EELD project was the external evaluation conducted under the EELD program in August 2003. The major activity for this evaluation was the development of the annotation guidelines and the annotation of gold standard documents by the three EE contractors for the EELD program between May and August 2003.

One of the key factors in developing the annotation guidelines is to develop an ontology that both fits the corpus and is suitable for human annotation. The starting point was to take the ontology developed in the earlier year by intelligence analysts working with Russian Contract Killing documents. The development then consisted of cycles of trial annotations by all the groups resulting in examples for the annotation guidelines and changes in the ontology. Ben Rode from Cyc and Alexis Mitchell from LDC played key roles in the development of the ontology and representing the perspective of human annotation, respectively, and in the writing of the annotation guidelines. Weekly teleconferencing provided working communication among all the participants.

Each of the three participants then annotated training documents up until August 4. Then each site annotated approximately a third of the evaluation documents between August 7 and August 26. The evaluation took place August 29. The time frame for the ontology development and annotation was very compressed, and there were changes incorporated into the ontology up until the last week before the evaluation, based on the annotation experience.

During this time, CNLP linguistic analysts were primarily involved in the annotation effort and had very little time for developing rules or for testing. The rules that were developed added some entity relations that had not been present in the CNLP ontology already, but some relations remained not covered due to lack of time.

The remaining CNLP development for the evaluation consisted of writing programs to prepare correctly formatted output and using TBL, as discussed in section III.a, to transform extractions using the CNLP ontology to those using the EE evaluation ontology as it was developed. The main difficulty in preparing the output format was in accurately representing text by means of character offsets in the original text. The eQuery extraction module is based on a document processing system, called TextTagger, that is token oriented and ignores white space between tokens. In order to produce character offset output, the text output is used to “rediscover” the token offsets, and a significant amount of inaccuracy was introduced into the output. A new document processing system, TextTagger 1.5, was under development at that time, but was not ready for the August evaluation. (This system is described in the next section.)

For the reasons described above, the CNLP score in the August evaluation was lower than the other participants, and it was felt this did not represent the capabilities of the

CNLP system. In October 2003, an analysis of the sources of errors was carried out. At this time, some small improvements were made, some additional training with TBL for EELD types was done, and some cleanups of duplicate relations and other small bug fixes were implemented. This resulted in some improvements in the score.

The results of the analysis showed that some relations were indeed missed due to insufficient development time, and other relations were missed because of the genre's sentence complexity, as described in section III.c above. But the main source of error was the inaccuracy of the character offsets generated to represent the EELD output from the text strings of the TextTagger output. After the completion of TextTagger 1.5 with a new internal representation to retain character offsets, we reran the EELD scoring with significant improvement as shown below.

Evaluation – August 29, 2003	45.8
Small improvements – October 2003	50.9
TextTagger 1.5 – early February 2004 correctly representing character offsets	57.8
Compared to human performance	79.9

Currently, the TextTagger system is undergoing additional improvements using a set of tools that allow the analysts to automatically find duplication and errors in the rulesets. The scoring will be rerun when this ruleset cleanup is complete.

One significant aspect of the EELD evaluation is that while all three contractors participated in the annotation and development of the guidelines, the analysis by GITI showed that interannotator agreement scores between the sites were lower between CNLP and the other two sites, than between the other two sites themselves. This demonstrates a difference in the understanding of the guidelines, that affected the development of the system, and hence the score.

The difficulty in understanding the guidelines is that they are based on complex human behavior and language, and are necessarily open to interpretation. (This is true for all annotation at this level of complexity.) One of the issues subject to interpretation is that even when trying to annotate relations that are explicitly stated in the text, there may be quite a few relations present between the entities in the text. The annotator is instructed to select only one relation present between those entities, based on string proximity. For example, in the phrase,

Margaret Williams , director of the WWF 's Bering Sea program, the following relations were annotated in the gold standard

hasLeaders (*WWF's Bering Sea program, director*)
subOrganization (*WWF, WWF's Bering sea program*)

In fact, on this example, the CNLP system, whose extractions are not based on term proximity, produced the following additional relations

affiliatedWith (*Margaret Williams, WWF*)
employees (*WWF, director*)

While both of these relations are correct, both in fact and in their expression as EELD constructs, they were scored incorrect, as "false alarms", since they were not in the gold standard. This illustrates a difference in interpretation and system implementation: the CNLP system tries to find all relations between a person and an organization, while the guidelines for the gold standard arbitrarily disallowed these relations in an attempt to manage the human annotation task.

V. System Development and Delivery

The eQuery Extraction Module is based on an underlying document processing system, called TextTagger. The original TextTagger versions included hardwired phases for

tokenization, POS tagging, sentence detection,
stemming, contractions, non-compositional phrases,
bracketing (temporal references, numeric concepts, named entities, noun phrases),
categorization,
generic relation extraction for entities and events, temporal relations,
entity coreference, and domain transformations

An additional phase for event coreference occurs in the system as a post-document processing phase that uses the frame logic inference system on extraction. The internal text representation was based on tokens, ignoring white space. Extractions were represented out-of-line in a table data structure.

The TextTagger document processing system was substantially rewritten between January 2003 and January 2004 to improve performance, address the character offset problem, and to make the system more flexible and robust.

One of the most important design goals was to improve the performance of the extraction tables. The representation of the tables for large documents was not scalable in the original implementation. A new data structure was designed for this that included additional hashing accesses to the table to improve performance. Additionally, there were some improvements made to the accuracy of the representation of heads and extents of overlapping noun phrases. This effort was very successful, as the performance figures show below.

Another design goal was to improve the accuracy and modularity of the POS (part-of-speech) tagging system. In the original TextTagger, a version of the application of Brill's POS tagging rules was implemented with considerable speedups, based on the POS tagging rules for text. The POS tagger was successfully rewritten to apply the more general POS tagging rules, with some sacrifice of speed.

For evaluations such as ACE and EELD, it is necessary to have an underlying text representation so that the tokens retain character offset information from the original document. Such a representation was built and carried out throughout the TextTagger rule phases and the extraction table.

Finally, the design goals were to include additional object-oriented aspects to TextTagger. The first was to make the TextTagger phases more modular to support "plug-and-play" modules at the phase level, for example, so that a different POS tagger or a statistical extraction phase could be inserted to replace a current rule-based phase. The second design goal was to create token objects and to replace the current rule-matching algorithm based on Perl regular expressions with a general algorithm based on object-oriented expressions. Such a system would be more easily ported to C++ or Java in the future. The first part of this design was completed and considerable work was done on the second part. But the general object-oriented rule matching was considerably slower than the Perl based rule-matching, and it was decided to leave the Perl rule-matching in place.

The performance testing of the resulting TextTagger 1.5 system showed a significant improvement, due primarily to the new extraction table representation. In particular, a performance test of files of the EELD evaluation document set is reported here. The system was run with all EELD extraction phases on a Windows XP machine with a 1.8Gh processor and 512Mb memory, the minimum that is recommended. The files were sorted by size and the performance reported by throughput, where larger numbers are better.

File size	TextTagger 1.0	TextTagger 1.5
Small (2-5 KB)	.07 kb/sec	.25kb/sec
Medium (6-9 KB)	.04	.22
Large (10-12 KB)	.03	.22

Figure 17. Performance improvements for TextTagger document processing

The new version of TextTagger 1.5 will also support a number of tools that will enable better statistical and rule development. Current tools under development include an automatic evaluation system from gold standard data and tools to analyze and organize rule systems to better streamline duplicate functionality and to automatically identify potential sources of error.

For the EELD program, the final software delivery of eQuery Extraction Module includes TextTagger 1.5. The module includes a parameter to allow the selection of the EELD evaluation ontology or the CNLP taxonomy. It also includes a parameter to select one of two output formats: the first includes an extraction for every entity and event mention to support direct comparisons from the text, and the second coalesces the entity and event mentions by coreference to support visualization and the filling of databases with entities.

The final software deliverable was made in March 2004.

VI. Transitions

This is the total list of transitions of the eQuery Extraction Module during the EELD program.

Saffron Tech – 3/02 – sent sample data for domain of hydro-electric power production to run through our EE software; we sent back the annotated sample set.

Hicks & Associates, Inc. – 4/02 – a SAIC subsidiary – a subcontractor in their INSCOM Proposal for advanced IE.

Raytheon 1 – Reston, VA – 2/02 demo’d EE system & licensed a research version.
Raytheon 2 – Lanham, MD – 3/02 demo’d EE system & licensed a research version.

Global Matrix – 3/02 – provided system overview and an analysis of QA data for use in dialogue clarification system.

Veridian (former MRJ division), 2002 – licensed eQuery Extraction Module.

EELD contractors, 2002 – delivered to GITI, SRI, Metron, USC, and Alphatech.

U.S. Army, Fort Huachucha, April 16, 2003. Delivered eQuery Evidence Extraction module with terrorist specialization rule set and eQuery Loader for CrimeLink Visualizer.

TIA, June 27, 2003, First demo’d & evaluated at Hicks & Associates, attended TIA developers’ workshop on July 10-11, delivered software in August.

Demo by AFRL with 21st Century, March 2003, Processed illegal drug activity news reports in eQuery, used output to generate ‘ground truth’ graph in TMODS and to display police-arrest-person instances.

CHI, February 2003 through May 2003, interchange of documents and results.

NYU, summer 2003, tagged newsfeed with entities, events, relations, NYU runs their relational learners on it, produces a typed relationship graph as input to pattern learning algorithms.

Syracuse Research Corp – September 2002 through July 2003, research collaboration has produced LiVIA, a two stage retrieval system, uses eQuery document-processing in terrorist and financial domains, currently being demo’d by SRC to various offices.

MySentient – August 2003, will use eQuery query-processing and document- processing modules for eLearning & eTraining products.

EELD contractors, 2004 – delivered to GITI, 21st Century, and Alphatech.

VII. Papers and Presentations

The following papers and presentations give further background and details on the research reported on in this report.

Representing Textual Content in a Generic Extraction Model, Nancy McCracken, AAAI Spring Symposium, 2001. (Supported by the EELD Seedling Project)

Transformation Based Learning for Specialization of Generic Event Extractions, Mary D. Taffet, Nancy J. McCracken, Eileen E. Allen, Elizabeth D. Liddy, CNLP technical report, 2002. (Supported by EELD)

Certainty Categorization Model, Rubin, Kando, and Liddy, AAAI Spring Symposium, 2004. (Supported by NSF)

Time-Bound: Capturing Temporal Information in Natural Language Texts, Svetlana Symonenko, Nancy McCracken and Elizabeth D. Liddy, Submitted to COLING 2004. (Supported by EELD)

Liddy, E.D. (2003). Natural Language Processing for Text Extraction Applications. Keynote Speaker. Thomson 8th Annual Text Summit. Minneapolis, MN. October 8, 2003.

Liddy, E.D. (2002). Specializing Evidence Extraction Using Transformation Based Learning. TIDES Annual Principal Investigators Meeting. Santa Monica, CA., July 25, 2002. http://www.cnlp.org/presentations/present.asp?show=conference.

Liddy, E.D. (2002). Advanced NLP-Based Information Extraction. Fusion-2 Workshop. Rome Lab, Utica, NY, June 26, 2002.

VIII. References

Allen, James F., *Maintaining Knowledge about Temporal Intervals*, CACM, Nov. 1983.

ARDA Summer Workshop on Graphical Annotation Toolkit for TimeML. Final report. http://nrrc.mitre.org/NRRC/TangoFinalReport.pdf

Baker, Collin F., Fillmore, Charles J., & Lowe, John B.(1998). The Berkeley FrameNet Project. Proceedings of the COLING-ACL, Montreal, Canada.

Brill, E. (1993). A corpus-based approach to language learning (Ph.D. Thesis). Philadelphia, PA: Department of Computer and Information Science, University of Pennsylvania.
Available at: http://www.cs.jhu.edu/~brill/dissertation.ps

Brill, E. (1994). Some advances in transformation-based part of speech tagging. Twelfth National Conference on Artificial Intelligence (AAAI-94) .
Available at: http://www.cs.jhu.edu/~brill/TAGGING_ADVANCES.ps

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics.
Available at: http://www.cs.jhu.edu/~brill/CompLing95.ps

Brill, E., & Resnik, P. (1994). A rule-based approach to prepositional phrase attachment disambiguation. COLING 1994 .
Available at: http://www.cs.jhu.edu/~brill/pp-attachment.ps

Collins, Michael, (1996). A New Statistical Parser Based on Bigram Lexical Dependencies. Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, ACL 1996, pages 184-191.

Cook, Walter A..(1998). Case Grammar Applied, Summer Institute
of Linguistics, Inc., 1998.

Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G. (2004). *TIDES - 2003 Standard for the Annotation of Temporal Expressions*.
http://www.mitre.org/work/tech_papers/tech_papers_04/ferro_tides/index.html

Ferro, L., Vilain, M., & Yeh, A. (1999). Learning transformation rules to find grammatical relations. Computational Natural Language Learning: A workshop at the 9th Conf. of the European Chapter of the Association for Computational Linguistics .

Filatova, E. & E.Hovy. Assigning Time-Stamps to Event-Clauses. ACL-2001 Proceedings.

Fillmore, Charles J. (1968). The Case for Case. In Emmon Bach and Robert Harms (eds.), Universals in Linguistic Theory. New York, Holt, Rinehart, and Winston, 1968. pp.1-88.

Gardner, W. (1995). On the reliability of sequential data: measurement, meaning, and correction. *In John M. Gottman (Ed.), The analysis of change.*Mahwah, N.J.: Erlbaum.

Gerber, L., Ferro, L, Mani, I, Sundheim, B, Wilson, G., and Kozierok, R (2002). Annotating Temporal Information: From Theory to Practice. In Proceedings of the 2002 Conference on Human *Language Technology*. San Diego, CA, 2002, 226-230.

Gildea, D., & Hockenmaier, J. (2003). Identifying semantic roles using combinatory categorial grammar. 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003) Association for Computational Linguistics.
Available at: http://www.cis.upenn.edu/~dgildea/gildea-emnlp03.pdf

Gildea, D., & Palmer, M. (2002). The necessity of parsing for predicate argument recognition. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02) (pp. 239-246). Association for Computational Linguistics.
Available at: http://www.aclweb.org/anthology/P02-1031.pdf

Halliday, M.A.K. & Hasan, R. (1976). Cohesion in English. London, Longmans.

Hobbs J., Pustejovsky J. *Annotating and Reasoning about Time and Events*. (2002). AAAI.

Hobbs, J., Stickel, M., Appelt, D., & Martin,P. (1993). Interpretation as Abduction. Artificial Intelligence, 63, 1993, pages 69-142.

Hockenmaier, J., & Brew, C. (1998). Error-driven learning of Chinese word segmentation. 12th Pacific Conference of Language and Information (pp. 218-229). Singapore: Chinese and Oriental Languages Processing Society.
Available at: http://www.ltg.ed.ac.uk/~chrisbr/papers/hockenmaier-colips98.1/

Iwanska, L. *Natural (Language) Temporal Logic: Reasoning About Absolute and Relative Time*. 1996.

Iwanska, Lucja M. (2000). Natural Language is a Powerful Language Representation System: the UNO Model, Natural Language Processing and Knowledge Representation, Chapter 1, edited by Iwanska and Shapiro, American Association for Artificial Intelligence, 2000.

Katzer, J., Bonzi, S., & Liddy, E.D. (1986a). The Effects of Anaphoric Resolution on Retrieval Performance: Preliminary Findings. Proceedings of the American Society for Information Science Annual Meeting, Vol. 23, 1986, 118-22.

Katzer, J., Bonzi, S. & Liddy, E.D. (1986b). Impact of Anaphoric Resolution in Information Retrieval. Final Report, National Science Foundation, July, 1986.

Paul Kingsbury and Martha Palmer. From Treebank to Propbank, 2002. In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas, Spain.

Landis, J. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, (33), 159-174.

Lappin, Shalom and Leass, Herbert. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535-561, 1994.

Liddy, E. D. (1990). Anaphora in natural language processing and information retrieval. Information Processing and Management. 26:1, 39-52.

Mani, I., Schiffman, B., Zhang, J. (2003). *Inferring Temporal Ordering of Events in News*.

Merriam-Webster Online Dictionary, http://www.m-w.com/ Accessed, January 30, 2004.
Palmer, D. P. (1997). A trainable rule-based algorithm for word segmentation. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL97) Association for Computational Linguistics.
Available at: http://ssli.ee.washington.edu/ssli/people/palmer/papers/acl97.ps

Ramshaw, L. A., & Marcus, M. P. (1996). Exploring the nature of transformation-based learning. In J. L. Klavens, & P. Resnik (Eds.), The balancing act: Combining symbolic and statistical approaches to language (pp. 135-156). Cambridge, MA: MIT Press.

Reichenbach, H, (1947). *Elements of Symbolic Logic*. Macmillan, London.

Rubin, V. L., Kando, N., and Liddy, E. D. (2004). Certainty Categorization Model, AAAI Spring Symposium on Attitude and Affect in Texts, 22-24 March 2004, Stanford, Palo Alto, CA. (A copy is attached to this report).

Satta, G., & Brill, E. (1996). Efficient transformation-based parsing. ACL 1996 Association for Computational Linguistics.
Available at: http://www.cs.jhu.edu/~brill/Eff_Pars.ps

Schilder & Habel. *From Temporal Expression to Temporal Information: Semantic Tagging of News Messages*. ACL-2001 Proceedings.

Sowa, J. F. (1984). Conceptual Structures, Information Processing in Mind and Machine, Addison-Wesley 1984.

Sowa, John F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole 2000.

Sundheim, B. *Association of Absolute Times with Events: Current Status and Future Changes*. July, 2002.

Wiebe, J., O'Hara, T.P., Ohrstrom-Sandgren, T., and McKeever, K.J. (1998). *An Empirical Approach to Temporal Reference Resolution*. Journal of Artificial Intelligence Research, vol.9, 247-293.

Wilson, G., and I. Mani, *Robust Temporal Processing of News*, ACL 2000. http://citeseer.nj.nec.com/mani00robust.html

Weischedel, R., Aone, C., Knoblock, C., Liddy, E., Mitchell, A., Rode, B., and Silk, B. *Data Preparation for Link Discovery: Building the Evidence Data Base*. Forthcoming.

van Rijsbergen, C.J. 1979. *Information Retrieval.* Butterworths, London

Vilain, M., & Palmer, D. P. (1996). Transformation-based bracketing: Fast algorithms and experimental results. Proceedings of Workshop on Robust Parsing (at ESSLLI-96) . Available at: http://ssli.ee.washington.edu/ssli/people/palmer/papers/esslli96.ps

Yeh, A. S., & Vilain, M. B. (1998). Some properties of preposition and subordinate conjunction attachments. 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL '98) Montreal, Canada: Association for Computational Linguistics. Available at: http://xxx.lanl.gov/ps/cmp-lg/9808007

Appendix A: CNLP Extractions and Taxonomy

March 1, 2004

CNLP extracts entities (which include named entities) and events from text documents and represents them as objects in extraction tables. These entities and events are put into frames with modifying slots. One of the primary slots is the “type” slot, which gives the type of the entity from the category hierarchy. Some slots can be viewed as attributes, while others are viewed as relations, primarily if their value is another entity. The attributes and relations are dynamically assigned as appropriate to entities and events during the extraction process.

The extracted entities include proper noun phrases and compositional noun phrases, except those denoting numeric concepts, which are treated separately. The generically extracted events are almost all verbs, particularly those denoting actions, but not including those which denote states of being, where the information is put into an attribute or relation instead.

Extractions, including entities and events, can be put together by equivalence groups. The types of equivalence represented are name variance, entity coreference and event coreference.

Category Hierarchy for Entities

Listed below is the hierarchy of categories that our linguistic analysts use to categorize entities. Strictly speaking, this hierarchy is a taxonomy, not an ontology, since we have no formal axioms that describe the relation between categories. On this copy of the taxonomy, each line lists the name of the category, followed by the abbreviation that would appear in the output.

0 Non-Human Living Things	
0.1 Animals	[an]
0.2 Plants	[plt]
1 Human Living Things	
1.0 People	[per]
1.1 Titles / Positions	[ti]
1.1.1 Honorifics	[honr]
1.1.2 Roles	[rol]
1.1.3 Military Ranks	[rnk]
1.2 Groups	[grp]
1.2.1 Organizations	[org]
1.2.1.1 Government Orgs	[govorg]
1.2.1.1.1 Courts	[crt]
1.2.1.1.2 Lawmaking groups	[lawgrp]
1.2.1.1.3 Military	[milgrp]
1.2.1.2 Terrorist Groups	[ter]

1.2.1.3 Military divisions	[milgrp]
1.2.1.20 Organizational subdivisions	[orgdiv]
1.2.2 Companies	[co]
1.2.2.20 company subdivisions (departments)	[codiv]
1.2.3 Religion	[rel]
1.2.4 Sports Teams	[sptm]
2 Thought, Communication and Communication Channels	[tht]
2.1 Processes	[proc]
2.2 Media	[med]
2.2.1 Documents	[doc]
2.2.1.1 Laws & Legal Cases	[law]
2.2.1.2 Forms	
2.2.1.3 Newspapers	[nwp]
2.2.1.4 Journals and magazines	[jrnl]
2.2.2 Videotapes/Movies	[vid]
2.2.3 Books	[book]
2.2.4 Internet	[inet]
2.2.4.1 URLs	[url]
2.2.4.2 e-mail	[e-m]
3 Buildings & Structures	[bldg]
3.1 Specialized Facilities	[spfac]
3.1.1 Educational institutions	[edu]
3.1.2 Hospitals and Clinics	[hosp]
3.1.3 Laboratories	[lab]
3.1.4 Museums	[mus]
3.1.5 Arenas/Stadiums	[stad]
3.1.6 Hotel	[hotel]
3.2 Monuments	[monmt]
4 Substances, Materials, Objects, and Equipment	
4.1 Products	[prod]
4.2 Weapons System	[weap]
4.2.1 Weapons Of Mass Destruction	[wmd]
4.2.1.1 Biological Weapons	[bweap]
4.2.1.2 Chemical Weapons	[cweap]
4.2.1.3 Nuclear Weapons	[nweap]
4.2.2 Bombs	[bomb]
4.2.3 Missiles	[mssl]
4.2.4 Rockets	[rckt]
4.2.5 Shells	[shel]
4.2.6 Mines	[min]
4.3 Vehicles	[veh]
4.3.1 Aircraft	[airc]

4.3.1.1 Commercial aircraft	[comairc]
4.3.1.2 Military aircraft	milairc]
4.3.2 Land Vehicles	[landveh]
4.3.2.1 Cars	[car]
4.3.2.1 Trains	[train]
4.3.3 Water Vehicles	[watveh]
4.3.4 Spacecraft	[space]
5 Science, Technology, and Industry	[sci]
5.1 Science Processes	[sciproc]
5.2 Software	[sftw]
5.3 hardware and Equipment	[hrdw]
5.4 Systems	[syst]
5.5 Biological / Chemical	[biochem]
5.5.1 biochemical Processes	[bcproc]
5.5.2 Medical	[med]
5.5.2.1 Diseases, Ailments	[dis]
5.5.2.2 Medicine, Drugs	[drug]
5.5.2.3 Medicinal Plants	[medpl]
5.5.2.4 medical Processes	[medproc]
5.5.2.5 Medical Specialties	[medspec]
5.5.3 Chemical Elements	[elmt]
5.6 Space / Astronomy / Physics	[astphy]
5.7.1 astronomic and physics Processes	[approc]
5.7.2 Stars	[star]
5.7.3 Constellations	[cstll]
5.7.4 Planet	[planet]
6 Social Sciences - Education / Government / Politics	[soc]
6.1 social sciences Processes	[ssproc]
6.2 Government programs	[govprog]
7 Numbers and Measurement	[meas]
7.1 Number	[numb]
7.1.1 phone	[phone]
7.1.2 SSN	[ssn]
7.2 Measures	[meas]
7.2.1 Weight	[wt]
7.2.2 Distance	[dist]
7.2.4 Volume	[flvol]
7.2.5 Money (Currencies)	[money]
7.2.5.1 Price	[price]
7.2.6 Memory	[memor]
7.2.7 Age	[age]
7.2.8 Rate	[rate]
7.2.9 Percent, ratio	[ratio]

7.2.10 Pressure	[prs]
7.2.11 Electricity	[elec]
7.2.12 Power	[power]
7.2.13 Frequency	[freq]
7.2.14 Area	[area]
8 Business and Commerce	[bus]
9 Entertainment	[enter]
9.1 Broadway Shows	[brdway]
9.2 TV Shows	[tv]
9.3 Fictional characters	[fict]
10 Transportation & Distribution	[trans]
(to include such things as troop movements and distribution of physical items)	
11 Geography / Location (miscellaneous)	[geo]
11.1 Continent	[cont]
11.1.1 Region	[reg]
11.1.2 Country	[cntry]
11.1.2.1 State	[st]
11.1.2.2 Province	[prov]
11.1.2.3 Prefecture	[pref]
11.1.2.4 Arab Country	[arcntry]
11.1.2.5 MiddleEastern Country	[mideast]
11.1.2.6 Soviet Republic	[sovcntry]
11.1.3 City, Town, Village	[city]
11.1.3.1 Address	[addr]
11.1.3.2 Highway	[hway]
11.1.4 Body of water	[water]
11.1.4.1 Ocean	[ocn]
11.1.4.2 Sea	[sea]
11.1.4.3 River	[rvr]
11.1.4.4 Lake	[lake]
11.1.4.5 Gulf	[gulf]
11.1.5 Island	[isl]
11.1.6 Park	[park]
11.1.7 Mountain	[mtn]
11.1.8 Desert	[des]
11.1.9. Beach	[bch]
11.1.10 Forest	[forest]
11.2 Adjectival geographic names (eg Russian)	[geoadj]
11.3 Language not also a geographic name	[lang]
12 Time	[time]

12.1 Clock Time	[clock_time]
12.2 Date	[date]
12.3 Minute(s)	[minute]
12.4 Hour	[hour]
12.5 Year(s)	[year]
12.6 Day(s)	[day]
12.7 Week(s)	[week]
12.8 Weekend	[weekend]
12.9 Month(s)	[month]
12.10 Season	[winter, summer, fall, spring]
12.11 Fiscal year	[fiscal_year]
12.12 Decade(s)	[decade]
12.13 Century	[century]
12.14 Historical period	[hist]
13 Named Event	[nevnt]
13.1 Special Event	[specevnt]
13.1.1 Holiday	[hol]
13.2 Weather Event	[weath]
13.3 Military Event	[milevnt]
13.3.1 Military Operation	[milop]
13.3.2 War	[war]
13.4 Flight	[flight]
13.5 Sporting Event	[sprt]
13.6 Political Event	[polevt]
14 General and Abstract Terms	[genabs]
15 Tests and Measures	[test]
16 Awards, Prizes and Honors	[awrd]
17 Unknown	[inknown]

Attributes of Entities

The slotnames that we use for entities are sorted into Attributes and Relations, where the difference is that the value of a relation is another entity, and the value of an attribute is something like a date or a string. In our extraction table, we don't make this distinction syntactically, all of these are listed as slotnames under the entity.

First we have attributes that identify the text phrase and its parts. These concepts are illustrated by the phrase "rundown flight school".
text (the head of the phrase – "school")

characteristic (adjectives describing the head – e.g. “rundown”)
kind (nouns describing subtype of the head – e.g. “flight”)
extent (the whole phrase of the entity mention – “rundown flight school”)

Here are the rest of the attributes, except for temporal and certainty attributes, which are described later:

type (category in the hierarchy - equivalent to Cyc isa)
acronym
age
aka (for alias, former name, also known as, nickname)
cost
distance
duration (length of time not expressed by specific dates, months, years
Eg five hours, many weeks, a few years)
measure (the broader term to capture numeric descriptions of things)
point-in-time (specific date or slice of time anchored to the timeline; Eg Monday, early 1970s, a few years ago)
price
quantity (specific numeric)
title

Relations for Entities

affiliation (tight connection for people/organizations & org/org)
amount (non-numeric)
area (fuzzy location, near)
associated (general connections not one of the more specific ones)
content
employer
geographic-affiliation
headquarters
isa (description - Example: “key figure in the Democratic party”)
leader
location
material
member (only for membership – per/group ; org/group)
origin
owner
part-of
relative (personal relatives such as mother, brother, etc.)
residence
responsible-for (things)
value

Attributes of Generic Events

For the slotnames used with events, they are sorted into Attributes, Roles and Relations.

type (the category comes from a small separate event list described below)
body-part
cause
charge
condition (condition for the event to take place)
instrument
manner
material
method
negation (quantifies the event)
payment
possible (quantifies the likelihood of the event)
price
purpose
reason
result
weapon

Temporal Attributes

occurs
before
after
frequency
temp_qual (gives a quality of the event, such as possible or future, or negation)
date-of-birth
date-of-death

Temporal Relations between events

holds
concurrent
before
after
since
included_in

Roles of Generic Events

according-to (information source of the event)
agent
destination

experiencer (for animate entities)
location
object
participants
path   (via, by way of)
recipient
source

Certainty Extractions

Text may also have certainty information extracted. Each certainty frame is attached to a particular sentence and may contain one or more certainty extractions. Each certainty extraction has

ctclue – the certainty clue is the text in the sentence giving rise to the certainty
level – either high or low if this certainty is above or below normal
ptview – the source of the certainty